

UNIVERSIDAD CARLOS III DE MADRID
ESCUELA POLITÉCNICA SUPERIOR

GRADO EN INGENIERÍA INFORMÁTICA

SIMULACIÓN EPIDEMIOLÓGICA Y VISUALIZACIÓN



Universidad
Carlos III de Madrid
www.uc3m.es

Autor: David del Río Astorga

Tutor: David Expósito Singh

Fecha: 21 Junio de 2013

Resumen.

La epidemiología ha ganado gran importancia en la sociedad debido principalmente a la propagación del virus de la gripe A en el año 2009. Esta epidemia tuvo gran impacto social debido a su rápida expansión pasando a ser rápidamente una pandemia mundial y al temor resultante en la sociedad. Esto provoco una gran alarma social y una falta de medios para controlar la enfermedad por parte de las autoridades sanitarias. Por este motivo, se han desarrollado diversas herramientas que permitan realizar previsiones sobre la propagación de enfermedades con el fin de adoptar medidas capaces de reducir el impacto sobre la sociedad.

Dentro de estas herramientas se encuentran una gran cantidad de simuladores del virus de la gripe. Estos simuladores mediante diversos modelos matemáticos y sistemas de representación de las poblaciones son capaces de generar una previsión sobre la propagación de la enfermedad. Los simuladores epidemiológicos desarrollados se dividen principalmente en dos grupos según el modelo de enfermedad que implementen. Según el modelo de enfermedad los simuladores se clasifican en estocásticos, si son modelos probabilísticos, o deterministas, si son modelos deterministas que generalmente se basan en ecuaciones diferenciales.

En este trabajo se parte del simulador EpiGraph el cual implementa un modelo estocástico, capaz de simular el comportamiento de la enfermedad de la gripe, y un sistema de representación de la población mediante grafos basados en las redes sociales. Sin embargo, las previsiones resultantes del simulador son difíciles de analizar a simple vista y el sistema que genera modelos de la población en grafos mediante muestras de los grafos de las redes sociales puede mejorarse notablemente.

Por este motivo, en este proyecto, se buscara, por un lado, diseñar e implementar un algoritmo de muestreo de grafos para generar los modelos de población utilizados en el simulador capaz de mantener las propiedades del grafo original y, por otro lado, se buscara diseñar una aplicación que sea capaz de mostrar los resultados de las previsiones permitiendo un análisis completo con la menor dificultad posible.

Abstract.

Epidemiology has gained great importance in society mainly due to the spread of influenza virus in 2009. This epidemic had a great social impact because of its fast expansion becoming a global pandemic in a short time and the resulting fear in society. This provoked a great social alarm and uncertainty emerged in health authorities on how to control the disease. For this reason, has developed different tools to make predictions about the spread of disease in order to take measures that may reduce the impact on society.

Among these tools are a large amount of simulators of influenza virus. These simulators are able to generate a forecast of the spread of the disease using mathematical models and populations representing models. Epidemiological simulators developed are mainly divided into two groups according to the implemented disease model. According to the disease model, simulators are classified as stochastic, if implements probabilistic models, or deterministic, if implements deterministic models that are usually based on differential equations.

This project starts from the epi-graph simulator which implements a stochastic model, able to simulate the behaviour of influenza disease, and a system of representation of the population by means of graphs based on social networks. However, the resulting forecasts are difficult to analyse with the naked eye and the system that generates the population models on graphs using sample graphs of social networks can be significantly improved.

For this reason, this project will seek, on the one hand, to design and implement a graph-sampling algorithm to generate population models used in the simulator to keep the properties of the original graph and on the other hand, design an application able to display the results of the forecasts allowing a complete analysis easily.



Índice

Resumen.....	2
Abstract.....	3
1. Introducción.....	10
1.1. Motivations.....	10
1.2. Objectives.....	11
1.3. Document structure.....	11
2. Estado de la cuestión.....	13
2.1. Epidemiología.....	13
2.2. Modelos epidemiológicos.....	13
2.3. Simuladores epidemiológicos.....	15
2.4. Análisis de tecnologías.....	16
3. Arquitectura de la plataforma.....	20
4. Diseño de la solución.....	25
4.1. Algoritmo de muestreo de grafos.....	25
4.2. Visualización.....	30
5. Requisitos.....	38
5.1. Requisitos de usuario.....	38
5.1.1. Requisitos de usuario de capacidad.....	38
5.1.2. Requisitos de usuario de restricción.....	40
5.2. Casos de uso.....	40
5.3. Requisitos software.....	45
5.3.1. Requisitos software funcionales.....	46
5.3.2. Requisitos software no funcionales.....	48
5.4. Matriz de trazabilidad.....	49
6. Evaluación.....	50
6.1. Evaluación del algoritmo de muestreo.....	50
6.2. Focos de infección en distintas ciudades.....	62
6.2.1. Madrid.....	63
6.2.2. A Coruña.....	67
6.2.3. Barcelona.....	73
6.2.4. Leganés.....	76
6.3. Evaluación de distintos valores de R_0	80
6.5. Evaluación con focos compuestos.....	91
7. Planificación.....	101
8. Presupuesto.....	102
9. Conclusions and future work.....	105
9.1 General conclusions.....	105
9.2 Personal conclusions.....	105
9.3 Future research.....	106
10. Referencias.....	107

Anexo I: Manual de usuario. 109

Contenido. 109

Requisitos hardware y software. 109

Guía rápida de uso. 109



Índice de figuras

Figura 1: Modelo SIR.	14
Figura 2: Modelo SEIS.	14
Figura 3: Modelo SITR.	15
Figura 4: Representación de un grafo.	20
Figura 5: Estructura de los grafos.	22
Figura 6: Arquitectura del sistema.	24
Figura 7: Diagrama de clases.	31
Figura 8: Representación k-NN.	34
Figura 9: Visualización $1/\text{distancia}^2$	35
Figura 10: Visualización $1/\text{distancia}^3$	35
Figura 11: Visualización población/ distancia^3	36
Figura 12: Visualización raíz de población/ distancia^3	37
Figura 13: Diagrama de casos de uso.	41
Figura 14: Epidemia con poblacion 250 en 5 días con Escalado.	50
Figura 15: Epidemia con poblacion 250 en 5 días con Albatross.	51
Figura 16: Matriz de adyacencia de 250 individuos mediante Escalado.	51
Figura 17: Matriz de adyacencia de 250 individuos mediante Albatross.	52
Figura 18: Epidemia con poblacion 1000 en 5 días con Escalado.	52
Figura 19: Epidemia con poblacion 1000 en 5 días con Albatross.	53
Figura 20: Matriz de adyacencia de 1000 individuos mediante Escalado.	53
Figura 21: Matriz de adyacencia de 1000 individuos mediante Albatross.	54
Figura 22: Epidemia con poblacion 5000 en 5 días con Escalado.	54
Figura 23: Epidemia con poblacion 5000 en 5 días con Albatross.	55
Figura 24: Matriz de adyacencia de 5000 individuos mediante Escalado.	55
Figura 25: Matriz de adyacencia de 5000 individuos mediante Albatross.	56
Figura 26: Epidemia 5000 individuos.	56
Figura 27: No infectados 5000 individuos.	57
Figura 28: Epidemia 50000 individuos.	57
Figura 29: Topología Albatross.	58
Figura 30: Topología Escalado.	58
Figura 31: Crecimiento de conexiones.	59
Figura 32: Dispersión de matrices.	60
Figura 33: Complejidad temporal Escalado.	61
Figura 34: Complejidad temporal Albatross.	61
Figura 35: Complejidad espacial Escalado.	62
Figura 36: Complejidad espacial Albatross.	62
Figura 37: Gráfica de número de infectados Madrid.	64
Figura 38: Día 45 simulación de Madrid.	65
Figura 39: Día 63 simulación de Madrid.	65
Figura 40: Día 98 simulación de Madrid.	66
Figura 41: Día 119 simulación de Madrid.	66
Figura 42: Gráfica de número de infectados A Coruña.	67
Figura 43: Día 66 primera simulación A Coruña.	69
Figura 44: Día 79 primera simulación A Coruña.	69

Figura 45: Día 101 primera simulación A Coruña.	70
Figura 46: Día 141 primera simulación A Coruña.	70
Figura 47: Día 61 segunda simulación A Coruña.	71
Figura 48: Día 79 segunda simulación A Coruña.	71
Figura 49: Día 118 segunda simulación A Coruña.	72
Figura 50: Día 72 cuarta simulación A Coruña.	72
Figura 51: Día 95 tercera simulación A Coruña.	73
Figura 52: Gráfica de número de infectados Barcelona.	74
Figura 53: Día 47 simulación Barcelona.	75
Figura 54: Día 58 simulación Barcelona.	75
Figura 55: Día 76 simulación Barcelona.	76
Figura 56: Gráfica de número de infectados Leganés.	77
Figura 57: Día 27 simulación Leganés.	78
Figura 58: Día 40 primera simulación Leganés.	79
Figura 59: Día 40 tercera simulación Leganés.	79
Figura 60: Día 45 tercera simulación Leganés.	80
Figura 61: Epidemia con $R_0=1.5$	81
Figura 62: Día 45 simulación $R_0=1.5$	82
Figura 63: Día 63 simulación $R_0=1.5$	83
Figura 64: Día 98 simulación $R_0=1.5$	83
Figura 65: Epidemia con $R_0=3$	84
Figura 66: Día 22 simulación $R_0=3$	85
Figura 67: Día 30 simulación $R_0=3$	86
Figura 68: Epidemia con $R_0=0.5$	87
Figura 69: Epidemia con $R_0=1$	88
Figura 70: Día 62 simulación $R_0=1$	89
Figura 71: Día 91 simulación $R_0=1$	90
Figura 72: Día 113 simulación $R_0=1$	90
Figura 73: Gráfica infectados totales respecto a R_0	91
Figura 74: Epidemia con 1 infectado en cada foco.	92
Figura 75: Día 52 primera simulación con 1 infectado en varios focos.	93
Figura 76: Día 75 primera simulación con 1 infectado en varios focos.	94
Figura 77: Día 45 tercera simulación con 1 infectado en varios focos.	94
Figura 78: Día 59 tercera simulación con 1 infectado en varios focos.	95
Figura 79: Día 51 cuarta simulación con 1 infectado en varios focos.	95
Figura 80: Epidemia con 4 infectados en cada foco.	96
Figura 81: Día 30 primera simulación con 4 infectados en varios focos.	97
Figura 82: Día 56 primera simulación con 4 infectados en varios focos.	98
Figura 83: Día 47 segunda simulación con 4 infectados en varios focos.	99
Figura 84: Día 48 tercera simulación con 4 infectados en varios focos.	99
Figura 85: Día 34 cuarta simulación con 4 infectados en varios focos.	100
Figura 86: Diagrama de Gantt.	101



Índice de tablas

Tabla 1: Comparativa de tecnologías.	18
Tabla 2: Clase main.....	32
Tabla 3: Clase mapa.....	32
Tabla 4: Clase heatMap.....	33
Tabla 5: Clase posición.....	33
Tabla 6: Plantilla requisitos de usuario.	38
Tabla 7: RQ-001-CA.	38
Tabla 8: RQ-002-CA.	39
Tabla 9: RQ-003-CA.	39
Tabla 10: RQ-004-CA.	39
Tabla 11: RQ-005-CA.	39
Tabla 12: RQ-006-CA.	39
Tabla 13: RQ-007-CA.	39
Tabla 14: RQ-008-CA.	39
Tabla 15: RQ-009-CA.	40
Tabla 16: RQ-001-RE.....	40
Tabla 17: RQ-002-RE.....	40
Tabla 18: RQ-003-RE.....	40
Tabla 19: Plantilla casos de uso.....	41
Tabla 20: CU-001.	42
Tabla 21: CU-002.	42
Tabla 22: CU-003.	43
Tabla 23: CU-004.	43
Tabla 24: CU-005.	44
Tabla 25: CU-006.	44
Tabla 26: CU-007.	44
Tabla 27: CU-008.	45
Tabla 28: Plantilla requisitos software.	45
Tabla 29: RQ-SW-001-FU.....	46
Tabla 30: RQ-SW-002-FU.....	46
Tabla 31: RQ-SW-003-FU.....	46
Tabla 32: RQ-SW-004-FU.....	46
Tabla 33: RQ-SW-005-FU.....	46
Tabla 34: RQ-SW-006-FU.....	47
Tabla 36: RQ-SW-007-FU.....	47
Tabla 37: RQ-SW-008-FU.....	47
Tabla 38: RQ-SW-009-FU.....	47
Tabla 39: RQ-SW-010-FU.....	47
Tabla 40: RQ-SW-011-FU.....	47
Tabla 41: RQ-SW-001-NF.....	48
Tabla 42: RQ-SW-002-NF.....	48
Tabla 43: RQ-SW-003-NF.....	48
Tabla 44: RQ-SW-004-NF.....	48
Tabla 45: RQ-SW-005-NF.....	48

Tabla 46: RQ-SW-006-NF.....	48
Tabla 47: RQ-SW-007-NF.....	49
Tabla 48: Matriz de trazabilidad.....	49
Tabla 49: Resumen ejecuciones de Madrid.....	64
Tabla 50: Resumen ejecuciones de A Coruña.....	68
Tabla 51: Resumen ejecuciones de Barcelona.	74
Tabla 52: Resumen ejecuciones de Leganés.	77
Tabla 53: Resumen evaluación con $R0=1.5$	82
Tabla 54: Resumen evaluación con $R0=3$	85
Tabla 55: Resumen evaluación con $R0=1$	88
Tabla 56: Resumen de ejecuciones con 1 infectado por foco.	92
Tabla 57: Resumen de ejecuciones con 4 infectados por foco.	97
Tabla 58: Planificación.....	101
Tabla 59: Salarios recursos humanos.	102
Tabla 60: Presupuesto IRPF.....	102
Tabla 61: Presupuesto recursos humanos.	102
Tabla 62: Presupuesto equipamiento.	103
Tabla 63: Presupuesto de Software.	103
Tabla 64: Resumen del presupuesto.....	104

1. Introducción.

This section describes in a comprehensive manner the characteristics of the problem it seeks to provide a solution in the development of this work.

This project is part of EpiGraph. Epigraph is a simulator of the propagation of influenza. In this work seeks to make more realistic simulations of the behaviour of influenza and make predictions about the expansion of the disease in the Spanish state. Epigraph uses social networks graphs to model the contacts over individuals and implements a mathematical model to describe accurately the behaviour of the disease. More in detail, this simulator consists of two basic models: social model and disease model.

Social model

In epigraph are two parts for modelling society. The first part is the data model that represents connections between individuals of the simulated population. This data model is a graph that represents the population of the city. This graph represents individuals at the edges and the connections between individuals in the arches.

On the other hand within the social model incorporates a transport model between cities. This model distinguishes between short and long distance. This simulates accurately the connections between cities. In this way it simulates the effect of transport in the spread of the disease.

Disease model

This model is a set of states and transitions of each of the stages of an infectious disease: **Sano, Incubation, Infection, Immune, Hospitalized y Died**. The transitions between different states are determined by a set of parameters that determine transition probabilities as the value of R_0 .

This model is responsible for performing the simulation by applying the model that defines the spread of the disease. This model calculates the different probabilities of spread of the disease depending on the type of connection between individuals and the time span in which the simulation is. This will simulate the interaction between related individuals within different scopes of a person's life such as: work relations, family and leisure relationships.

This model also allows for variations on the scenario such as the including vaccinations or modify the parameters that define the disease.

1.1. Motivations

This project seeks to improve the functionality of epigraph in two directions: improving the accuracy of the simulations and improve the interface offered by the application.

On the one hand, due to be able to obtain more accurate predictions, it becomes necessary to improve the graph generation algorithm. This algorithm starting at connections contained in the graphs of social networks generates a smaller sample size that maintains existing properties in the network of real connections. Thus, the connections that are generated in the network used in the simulator will be more realistic and similar to urban social environment.

On the other hand the simulator provides just a plain text output difficult to analyze and check how it expands with the naked eye disease by the Spanish territory. For this reason it becomes necessary to develop a tool to visualize the output data in a graphical form to allow easy analysis of the forecasts made by the simulator. Using this tool, it would be possible to perform a much more complete analysis about the results of the simulation. This would allow a better understand the process of spreading nationwide.

1.2. Objectives.

According to the general approach outlined in the motivation, this project seeks to achieve the following objectives:

- The first objective is to implement a graph-sampling algorithm for epigraph epidemiological simulator, developed by ARCOS Department of UC3M, in order to improve the realism of the simulation and the connections between individuals in a population set. This algorithm will replace the current algorithm implemented by the simulator.
- The second objective is to develop a visual interface to represent the simulation results on a map. This way you could visually analyse the behaviour of the epidemic in Spain and how it spreads.
- The third objective is to perform a study to evaluate the effect on the spread of the disease using graphs generated by the new algorithm compared to the behaviour with current graphs.
- Finally, it seeks to perform an analysis of the disease behavior in different scenarios such as different initial foci of infection or changes in the value of the parameter that defines the degree of transmission of the disease (R_0).

1.3. Document structure.

This section describes the structure of the document giving a brief summary of its contents in order to facilitate reading.

- **Section 1: Introduction.** This section is intended to introduce the project content, placing it in the context and specifying the objectives.
- **Section 2: State of the question.** This section lists other implementations of simulators developed and evaluates different technologies for the implementation of the visual interface.
- **Section 3: Platform architecture.** This section defines the architecture of the system indicating the function of each module and sub-systems to be implemented in this project.
- **Section 4: Solution design.** This section describes the development of the project by defining a solution to the problem.
- **Section 5: Requirements.** This section describes the requirements defined for the visual interface design for the simulator results.



- **Section 6: Evaluation.** It describes and analyzes the results obtained for different scenarios.
- **Section 7: Planning.** This section defines the planning followed throughout the project development.
- **Section 8: Budget.** This section describes the calculation of costs to estimate the total cost of the project.
- **Section 9: Conclusion y future works.** This section provides the conclusions drawn from development of the project and possible future lines.

2. Estado de la cuestión.

2.1. Epidemiología.

La epidemiología es la rama de la ciencia que estudia las enfermedades existentes en la población humana. Uno de los casos de estudio de esta rama científica es el estudio de la propagación de enfermedades y las pandemias con el fin de poder realizar predicciones sobre estas enfermedades con el fin de minimizar los daños que puedan ocasionar sobre la población.

Las pandemias son un caso de estudio bastante importante, ya que, es un problema que afecta al conjunto de individuos de la sociedad. Actualmente éste es un campo de gran interés ya que existen grandes intereses en minimizar estos daños por los costes que producen a la sociedad.

Por este motivo se han diseñado una gran cantidad de modelos computacionales para simular la propagación de las enfermedades con el fin de controlar y responder ante enfermedades que puedan producir alarma social como ocurrió con la pandemia del año 2011 por el virus H1N1 conocido como la gripe A. Esta pandemia se convirtió en una gran alarma social debido a que esta enfermedad tenía una alta tasa de mortalidad y a la rápida propagación que tuvo convirtiéndose en una pandemia global.

Dentro de este contexto se sitúan los simuladores desarrollados para la propagación de la epidemia con el fin de predecir el comportamiento de la enfermedad y de esta manera establecer una estrategia con la cual enfrentarse a la enfermedad minimizando los riesgos.

2.2. Modelos epidemiológicos.

Con el fin de realizar predicciones sobre la propagación y el impacto de una enfermedad existen ciertos modelos como herramientas para realizar estos trabajos.

El modelo básico que implementan los simuladores es el modelo Susceptible-Infective-Recovered. Este modelo se basa en un pequeño modelo basado en estados, desarrollado por Kermack y McKendrick [14], que considera tres estados en los que puede encontrarse un individuo. De igual manera todos los individuos que conforman la población son considerados iguales en cuanto a como se enfrentan a la enfermedad. Dentro de este modelo se distinguen los siguientes estados:

- Susceptible: En este estado se encuentran los individuos que no han sufrido la enfermedad y que no poseen inmunidad a la misma, por lo que podrían ser infectados por la enfermedad.
- Infectado: En este estado se encuentran aquellos individuos que en un instante de tiempo se encuentran infectados por la enfermedad y son transmisores de la enfermedad a aquellos individuos en el estado de Susceptible.
- Eliminado: Este conjunto representa a aquellos individuos que han superado la infección y, por tanto, han desarrollado inmunidad a la enfermedad.

En la figura 1 se muestra el diagrama de flujo de los estados de este modelo en el cual los individuos parten del estado susceptible.



Figura 1: Modelo SIR.

Partiendo de este modelo se han desarrollado variantes para representar el comportamiento y propagación de las enfermedades que se basan en extensiones del sistema básico.

Algunos de estos modelos son el SEIR y el Sitr [15]:

El primero de estos modelos SEIR añade al modelo un nuevo estado, Exposed, el cual representa a aquellos infectados que se encuentran en el periodo de estado latente de la enfermedad lo que significa que están infectados pero no son contagiosos para el resto de individuos susceptibles. Igualmente existen diferentes variaciones en función de las enfermedades como SEIS el cual permite mejorar la precisión para enfermedades a las cuales los individuos infectados que superan la enfermedad no desarrollan inmunología hacia ella. En la figura 2 se muestra el diagrama de flujo de los estados de este modelo.

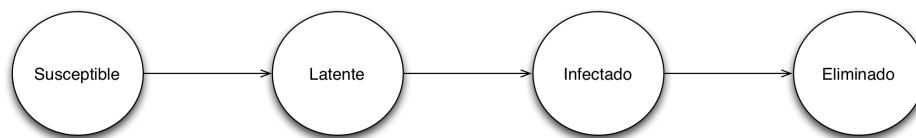


Figura 2: Modelo SEIS.

El modelo Sitr es un modelo el cual trata un nuevo estado, Treated, el cual representa el estado de aquellos individuos que son vacunados antes del inicio de la epidemia lo que implica que pasen a estar en un estado de protección ante la enfermedad al desarrollar el sistema inmune frente a la enfermedad. De esta manera se busca aumentar la precisión del modelo minimizando el porcentaje de individuos infectados a lo largo de la epidemia y adaptándolo al comportamiento real. En la figura 3 se representa el diagrama de estados de este modelo.

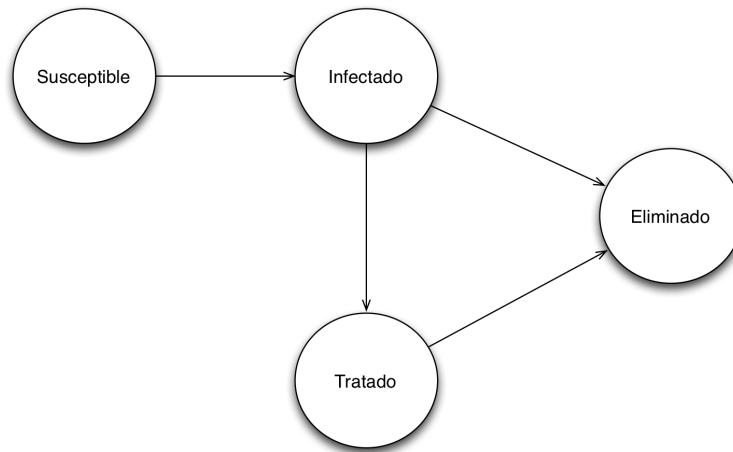


Figura 3: Modelo SITR.

2.3. Simuladores epidemiológicos.

Dentro de la simulación de epidemias se encuentran dos grandes grupos de simuladores: simuladores deterministas y simuladores estocásticos.

Los simuladores deterministas son aquellos que no se basan en procesos aleatorios o probabilísticos y que vienen definidos por ecuaciones diferenciales que determinan la enfermedad. Por este motivo los resultados de estas simulaciones es siempre el mismo ante un conjunto de condiciones y estados iniciales dados al sistema.

Los simuladores estocásticos son simuladores que se basan en probabilidad que se encuentran condicionados por variables aleatorias. Esto implica que los resultados para un mismo estado inicial den resultados distintos debido a la aleatoriedad.

Dentro de estos últimos simuladores se distinguen aquellos que se basan en modelos de contacto de las redes sociales. Estos simuladores basados en redes sociales permiten obtener resultados realistas debido a que la propagación de las enfermedades están estrechamente relacionadas con la estructura de las conexiones existentes dentro de la población.

Algunos simuladores estocásticos basados en redes sociales son los siguientes:

- **EpiFast** [11]: es un simulador que implementa el modelo de enfermedad SIR, comentado anteriormente, para simular la propagación de la epidemia. Las relaciones entre los individuos vienen dadas por estructuras basadas en redes sociales generadas aleatoriamente en las que no se incluye información demografía o geográfica. Es un sistema que a su vez está desarrollada en computación paralela con MPI que permite ejecutar el algoritmo en varios procesadores simultáneamente con el fin de minimizar el coste de tiempo.
- **FluTE** [12]: Este sistema está basado en el modelo basado en el individuo para el virus de la gripe. Un sistema basado en el individuo implica que los individuos se consideran agentes independientes. Este sistema construye la estructura de las conexiones a partir de datos extraídos de los censos de población. Sin embargo, las conexiones entre individuos se generan

aleatoriamente lo que implica un escaso realismo de las mismas. Este sistema al igual que el anterior implementa el sistema mediante un algoritmo paralelizado mediante MPI aunque se reduce mucho el rendimiento para comunidades de gran tamaño.

- **EpiSimdemics** [13]: Este sistema, al igual que el anterior, se basa en un modelo basado en el individuo. Este sistema implementa un algoritmo paralelo implementado en MPI que simula la propagación de enfermedades infecciosas en redes sociales generadas mediante información demográfica considerando los modelos de transporte. Este sistema es capaz de simular la infección en poblaciones de gran tamaño.
- **EpiGraph** [14]: Éste sistema es el sobre el que se trabajará a lo largo del desarrollo de este proyecto. Este es un sistema cuya red de conexiones esta generada basándose en las estructuras de conexión de redes sociales. Las conexiones de las poblaciones utilizadas en la simulación son generadas mediante un algoritmo de escalado de regla de tres. Este sistema implementa el sistema del simulador en programación paralela con MPI que es capaz de generar simulaciones de grandes poblaciones con un coste de recursos computacionales mínimo. Por otro lado el modelo epidemiológico que implementa este sistema es más complejo que los ya comentados, similar a una hibridación entre SIR y SITR, añadiendo nuevos estados como fallecido, y una conexión entre los estados más compleja, lo que permite representar con mayor precisión el comportamiento de la enfermedad.

2.4. Análisis de tecnologías.

Para el desarrollo del visualizador de los resultados generados por el simulador EpiGraph se han tenido en cuenta las siguientes tecnologías:

- HTML y JavaScript.
- Java.
- XCode.

Para evaluar cada una de las tecnologías de este análisis se tendrán en cuenta las siguientes características:

En primer lugar se tendrá en cuenta el coste como un factor limitante para el desarrollo del proyecto. Dentro de esta característica se incluyen tanto los costes de los equipos necesarios para el desarrollo como el coste de licencias.

Por otro lado se tendrán en cuenta tanto el número de referencias y tutoriales disponibles en la web sobre la tecnología, así como, los conocimientos sobre la tecnología de los que parte el equipo de desarrollo y la curva de aprendizaje de cada una de las tecnologías.

Otra característica que se valorará es la portabilidad de dicha tecnología a diferentes equipos con diferentes sistemas operativos. También se valorarán otras características como son el rendimiento.

Y, por ultimo, como factor más importante, la capacidad de utilizar mapas y localización por coordenadas dentro del mismo. Se analizarán las diferentes bibliotecas de mapas que se encuentren para cada una de las tecnologías y se escogerá aquella que permita desarrollar una aplicación con el mejor resultado posible.

HTML y JavaScript

Esta herramienta de desarrollo esta diseñada para aplicaciones web permitiendo ser ejecutadas en navegadores web que puedan ejecutar código HTML y JavaScript. Esto permite que las aplicaciones desarrolladas con esta herramienta se pueda garantizar que serán compatibles en gran cantidad de equipos, necesitando únicamente de un navegador Web compatible.

Por otro lado su coste es bastante bajo ya que no requiere de un equipo específico ni tiene necesidad de licencias para su desarrollo y publicación.

En cuanto a la curva de aprendizaje, se trata de un lenguaje bastante simple y se parten de un cierto conocimiento sobre el uso de esta tecnología, sobre la cual se ha trabajado durante la carrera, y de los mapas proporcionados por Google mediante la API v3 de JavaScript. Esto implica que el coste de aprendizaje para esta tecnología será relativamente bajo en comparación a las otras tecnologías consideradas. A esto se le añade que partiendo de estos conocimientos se pueda llegar a una solución con una mejor calidad.

Por ultimo, en cuanto a los mapas, se dispone de la biblioteca de GoogleMaps proporcionada por Google para JavaScript. Se trata de una biblioteca bastante completa que ofrece implementaciones para mapas de calor y geolocalización bastante sencillas de utilizar.

Los mapas de calor ganan gran importancia para este proyecto ya que permiten mostrar gráficamente sobre el mapa mediante un layout la información almacenada en una matriz que representa el estado de la enfermedad en los distintos puntos ubicados en una posición indicada como latitud y longitud. Esta implementación permite representar los datos procesados de la salida del simulador y representar tantos puntos como se definan en la matriz sobre el mapa mediante un código de colores (de verde a rojo) que representa el valor almacenado en cada posición de la matriz.

Java: Eclipse IDE

Esta herramienta está diseñada para realizar aplicaciones de escritorio compatibles en cualquier equipo que sea capaz de ejecutar Java. Java se trata de un sistema multiplataforma y, por este motivo, sus aplicaciones tienen una gran portabilidad.

Igualmente el coste para desarrollar aplicaciones en este lenguaje es bastante bajo en cuanto a dinero se refiere, ya que, se dispone de entornos de desarrollo como es Eclipse que no plantean ningún coste adicional y facilitan el desarrollo de las aplicaciones. Además se trata de un lenguaje que no requiere del pago de licencias y que pueden ser desarrolladas en cualquier equipo.

Por otro lado, existen gran cantidad de referencias y tutoriales en la web sobre este lenguaje que ofrecen gran ayuda para los desarrolladores. Además existen gran cantidad de bibliotecas públicas que pueden ser utilizadas para facilitar el desarrollo incorporando funcionalidades ya desarrolladas por la comunidad de desarrolladores de Java.

Sin embargo, este sistema, al necesitar de una máquina virtual, Java Virtual Machine (JVM) el rendimiento de estas aplicaciones se reduce bastante.

En cuanto a los mapas necesarios, al ser una aplicación de escritorio, GoogleMaps no permite su uso directo. Debido a esto se hace necesario recurrir a otras librerías como son los mapas proporcionados por TerraService o ArcGIS. Para el desarrollo de esta aplicación, la biblioteca más cercana a lo buscado se trata de la biblioteca proporcionada por Esri, ArcGIS Runtime SDK.

Sin embargo, en todas las posibles alternativas, la documentación existente y código de ayuda en la web es muy escasa, lo que implica una gran dificultad para el desarrollo de la aplicación. Además hay que contar con que se parte de un completo desconocimiento de estas librerías.

Objective C: XCode

La principal desventaja existente en este lenguaje es poca portabilidad existente. Debido a que este lenguaje es propio de Apple el desarrollo de una aplicación en este lenguaje queda limitado su uso a equipos Mac. Debido a que se busca desarrollar un sistema multiplataforma para visualizar los resultados, esta desventaja constituye una restricción muy importante desarrollo del proyecto.

Por otro lado, se tiene otro factor limitante en su coste. Aunque el entorno de desarrollo XCode es gratuito, se necesita de un equipo Mac para su desarrollo y el pago de la licencia de desarrollador para poder publicar la aplicación en la Mac App Store. El precio de la licencia de desarrollador de Apple es de 99 dólares al año.

Como ventajas se encuentra el gran rendimiento tanto en memoria como en uso de CPU ya que, al ser un superconjunto del lenguaje C no necesita de una máquina virtual como ocurre con Java.

En cuanto a los mapas, no se ha podido encontrar una librería para Objective-C. La posible solución es utilizar los mapas de Google a través de JavaScript mediante una librería de Objective-C que permita visualizar contenidos web dentro de la aplicación desarrollada. La librería ofrecida por Apple para llevar a cabo este cometido es *WebKit View* [9] que introduce dentro de la interfaz una vista de contenido web capaz de mostrar código HTML.

A continuación se muestra una tabla a modo de resumen de los resultados de este análisis:

	HTML y JavaScript	Java	XCode
Coste económico	Bajo	Bajo	Alto
Número de referencias	Alta	Media	Media
Curva de aprendizaje	Baja	Media	Alta
Portabilidad	Alta	Alta	Baja
Rendimiento	Bajo	Medio	Alto
Conexión	Necesaria	Necesaria	Necesaria
Mapas	GoogleMaps	ArcGIS	GoogleMaps

Tabla 1: Comparativa de tecnologías.

Tras realizar el análisis detallado anteriormente, se ha decidido desarrollar la aplicación en HTML y JavaScript debido a la gran cantidad de referencias encontradas tanto para la tecnología como para la biblioteca de mapas utilizada (GoogleMaps v3). Además, los altos conocimientos base de los que se dispone sobre esta tecnología. Esto se traduce en un coste de aprendizaje muy bajo y permite desarrollar una solución con mejores resultados.

Otras ventajas que se encuentran en esta tecnología es que puede ser ejecutada en cualquier equipo con un navegador web capaz de ejecutar HTML y JavaScript. Sin embargo, a pesar de las ventajas que presenta esta tecnología, existen algunas desventajas como son el bajo rendimiento y la necesidad de una conexión a la red para visualizar los mapas. Sin embargo, en el contexto de este proyecto, el rendimiento no es un factor crítico por lo que no se ha tomado en gran consideración.

Las otras posibles opciones analizadas se han descartado por diversos motivos. Realizar la aplicación en XCode para Mac se ha descartado principalmente por su coste y su baja portabilidad. En cuanto a usar Java para desarrollar la aplicación de escritorio ha sido descartado por la dificultad que plantea la incorporación de los mapas.

3. Arquitectura de la plataforma.

En esta sección se describe la arquitectura del sistema indicando los módulos que se van a implementar en el desarrollo del trabajo.

Este sistema consta de dos grandes componentes. Por un lado se encuentra el módulo de cómputo, este módulo es el propio EpiGraph. Este componente es el encargado de, por un lado, realizar las tareas previas para representar las diferentes poblaciones de la simulación y, posteriormente, aplica el proceso matemático que simula la propagación de la enfermedad dependiendo de los parámetros introducidos.

Por otro lado, se encuentra el módulo de visualización que permite mostrar los resultados del módulo anterior sobre un mapa para analizar de forma visual la propagación de la enfermedad.

3.1. Módulo de Cómputo.

Este módulo se divide en dos módulos más pequeños que concretan la funcionalidad del sistema. El primero de ellos es el generador de grafos que recibe como entrada los grafos proporcionados por redes sociales.

Un grafo es un conjunto de vértices o nodos interconectados entre ellos mediante aristas o arcos que representan las relaciones binarias entre los elementos del conjunto. Los arcos pueden tener distintas propiedades como es la dirección de la conexión que puede ir en una única dirección o en ambas. Igualmente pueden estar pesadas por un coste.

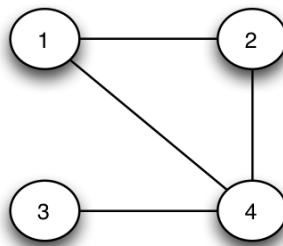


Figura 4: Representación de un grafo.

En la figura 4 se representa gráficamente un grafo de 4 nodos relacionados mediante una serie de aristas.

De la misma manera que se representa gráficamente los grafos pueden ser representados mediante una matriz en las que se representa en filas y columnas los nodos del grafo incluyendo el valor de la arista en las diferentes posiciones de la matriz. A continuación se muestra la representación matricial del grafo mostrado como ejemplo en la figura 4.

	1	2	3	4
1	0	1	0	1
2	1	0	0	1

3	0	0	0	1
4	1	1	1	0

En esta matriz se representa mediante unos las aristas del grafo que conectan cada nodo y con ceros aquellas conexiones que no existen en el grafo.

Dentro del sistema los nodos representan los diferentes individuos de la población y los arcos representan la existencia de conexiones entre ellos. En los grafos del sistema las aristas son bidireccionales, es decir, si A está conectado con B, B está conectado con A y no se encuentran asociados a un peso o coste.

Estos grafos se almacenan en el formato de matrices dispersas Compressed Sparse Column (CSC). Este formato almacena únicamente las entradas no nulas del grafo permitiendo así un gran ahorro de memoria. El sistema que sigue para dicho almacenamiento es el siguiente:

Se almacenan 2 vectores. El primero de ellos tiene una entrada por cada vértice del grafo en el que se indica mediante un número entero las conexiones existentes para ese vértice. El segundo vector almacena el vértice de destino de las aristas del grafo. Este sistema de almacenamiento para el grafo de ejemplo sería el siguiente:

El primer vector que indica el número de conexiones acumuladas de cada vértice del grafo. Es decir, el primer vértice tiene dos aristas y su valor será dos, el segundo vértice tiene dos también y su valor será dos por sus aristas más dos por las aristas del anterior, en total cuatro.

Vértices	1	2	3	4
Conexiones	2	4	5	8

El segundo vector almacena los vértices de destino de cada una de las aristas de tal manera que los dos primeros vértices destino serán del primer vértice, los dos siguientes del segundo y así sucesivamente.

Arista	1	2	3	4	5	6	7	8
Destino	2	4	1	4	4	1	2	3

De esta manera se reduce la cantidad de memoria necesaria para representar la información del grafo. En el caso de ejemplo se pasa de necesitar almacenar una matriz de 16 posiciones a dos vectores de en total 12 posiciones.

Los grafos de entrada utilizados en el sistema contienen las relaciones existentes entre los usuarios de dicha red. Estos datos recibidos como entrada son procesados para obtener un grafo de menor tamaño con el fin de facilitar la posterior simulación.

Los grafos de entrada de los que se dispone son extraídos de la red social Facebook y de la empresa energética Enron Corporation. Estos grafos contienen las conexiones existentes entre los individuos que conforman estas redes en el formato CSC comentado anteriormente.

Con estos grafos de entrada de gran tamaño mediante diferentes métodos se genera un grafo de salida de menor tamaño. Dentro de este grafo, los individuos se encuentran divididos en grupos de colectivos de diferentes tamaños que comparten un mismo sitio de trabajo y que, por tanto, mantienen una relación estrecha. En la figura 5 se muestra la estructura del grafo mostrando únicamente las conexiones de los grupos de individuos que conforman el grafo.

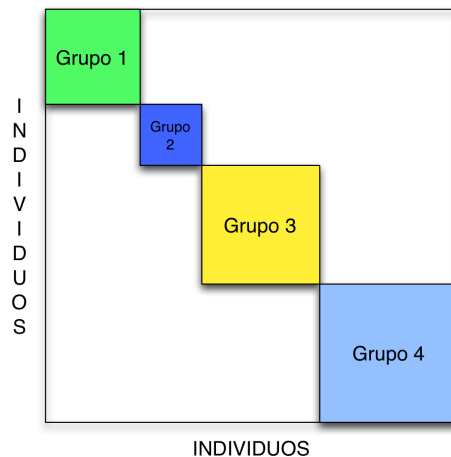


Figura 5: Estructura de los grafos.

Por otro lado se encuentran las conexiones existentes entre los diferentes grupos referentes a la familia y las amistades. Estas conexiones se diferencian de las anteriores en que se dan durante el tiempo de ocio de los individuos. En la siguiente grafica se muestra la estructura del grafo en la que únicamente se muestran las conexiones externas de los grupos.

El módulo del generador de grafos consta de varios submódulos de los cuales se trabajará sobre dos de ellos. El primero de estos módulos, el componente interno, es el encargado de generar las conexiones existentes dentro de los grupos que conforman el conjunto global de la población. El segundo módulo, el componente externo, se encarga de generar todas las conexiones existentes entre cada par de grupos.

Sobre estos submódulos se trabajará con el fin de generar grafos con propiedades de conectividad y distribución de las conexiones más similares a las existentes en un grafo de relaciones real.

Por otro lado, el sistema inicial consta de un segundo módulo, que es el propio simulador. Este módulo utiliza un modelo matemático que simula cómo se extiende la enfermedad dentro de las ciudades y otro modelo para simular el transporte entre ciudades. El modelo utilizado para la enfermedad se basa en un conjunto de estados en los que se encuentran los individuos y transitan de un estado a otro dependiendo de probabilidades. Los estados utilizados son: **Sano, Incubación, Infección, Inmune, Hospitalizado y Fallecido.**

Para el modelado del transporte se utiliza el modelo gravitacional de la teoría del transporte [16]. Este modelo esta basado en el modelo gravitacional utilizado en econometría. Este modelo establece una dependencia existente, en el flujo entre las comunidades , entre la distancia y el tamaño de las comunidades de residencia y trabajo. Este modelo calcula el flujo de individuos como:

$$C_{ij} = \theta \frac{P_i^{t1} P_j^{t2}}{d_{ij}^p}$$

Siendo C el flujo entre las ciudades i y j , P las poblaciones y θ una constante de proporción.

Las variables $p, t1$ y $t2$ son parámetros para modificar la dependencia entre las poblaciones y la distancia

De esta manera se puede modelar el transporte de trabajadores teniendo un descenso muy rápido en el número de transportes conforme aumenta la distancia hasta alcanzar unos valores de individuos que se transportan que prácticamente no dependen de la distancia. Igualmente los transportes entre ciudades cercanas son relativamente altos en el transporte de trabajadores. Este modelo permite simular con precisión el modelo de transporte.

Este módulo ofrece cómo salidas una serie de ficheros de texto que contienen los datos referentes a la predicción realizada por la simulación.

3.2. Módulo de visualización.

Partiendo de el módulo anterior, se busca desarrollar un nuevo módulo que, tomando como entrada la salida del proceso anterior, permita visualizar de forma gráfica la previsión resultante con el fin de favorecer el análisis de la propagación de la enfermedad entre las diferentes ciudades.

Este módulo procesa los datos de resultados de la simulación y muestra sobre un mapa el estado de la enfermedad en los diferentes tiempos de la simulación, permitiendo ver de forma gráfica su estado.

En la figura 6 se muestra de forma esquemática la arquitectura del sistema indicando las partes desarrolladas y modificadas en color azul. En esta figura se representan los diferentes módulos en cajas en las que se muestra el nombre y los diferentes elementos de entrada y salida de los módulos mostrados como texto plano. Dentro de cada gran módulo se muestran los diferentes submódulos que componen la arquitectura del sistema. En esta figura se puede seguir de una forma sencilla el flujo del sistema que se inicia recibiendo los grafos originales de las redes sociales hasta obtener un como ultima salida la visualización de los resultados de la simulación.

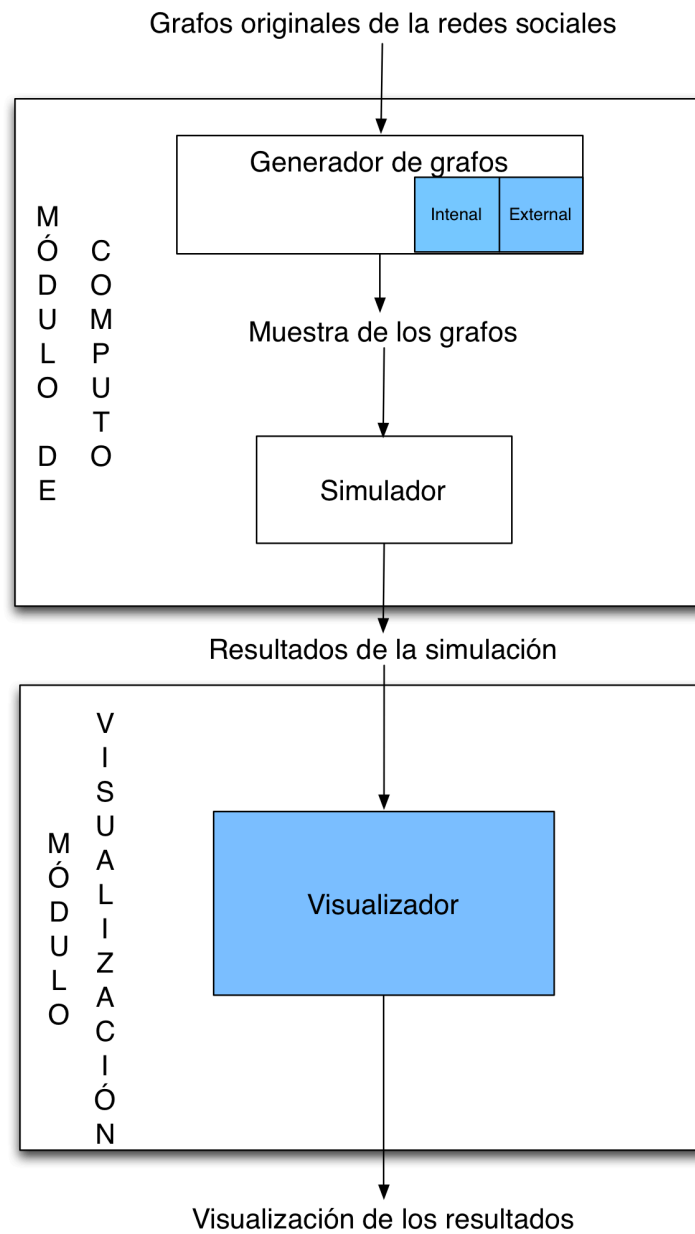


Figura 6: Arquitectura del sistema.

4. Diseño de la solución.

En esta sección se detalla el proceso realizado para el desarrollo del trabajo distinguiendo dos bloques. El primero de ellos describe el desarrollo para la implementación del algoritmo de muestreo de grafos y el segundo describe el desarrollo de la aplicación para visualizar sobre un mapa los resultados de las simulaciones.

4.1. Algoritmo de muestreo de grafos.

En el desarrollo de este primer bloque del proyecto se parte del simulador EpiGraph. Este simulador genera un grafo de un tamaño reducido a partir de grafos obtenidos de las redes sociales manteniendo las propiedades originales de estos últimos. Este grafo se genera dos etapas:

En la primera etapa se generan las conexiones existentes dentro de un mismo grupo de individuos mediante un algoritmo de escalado que toma una columna aleatoria del grafo de origen y mediante una regla de tres determina el número de conexiones entre los individuos de un grupo. Los grupos generados poseen características distintas dependiendo del tipo de conjunto que represente: trabajadores, estudiantes, amas de casa y ancianos.

A continuación se muestra el pseudocódigo del algoritmo que genera las conexiones del grafo.

```
Desde individuo = 0 hasta individuo = num_individuos_colectivo
  Hacer
    Seleccionar individuo del grafo origen al azar
    Obtener el número de conocidos del individuo del grafo
    origen(num_con)
    Mientras ( num_individuos_colectivo * umbral < num_con )
  Fin Desde
```

Este algoritmo se aplica para colectivo selecciona para cada individuo del colectivo un individuo aleatorio del grafo de origen. De este individuo seleccionado se obtiene el número de conexiones existentes y se comprueba si este individuo posee un número de conexiones mínimo que queda establecido como el numero de individuos totales por el valor umbral. Si se cumple se pasa al siguiente individuo, si no, se continúan seleccionando individuos.

Una vez seleccionado el individuo que cumple esta condición se escala el tamaño del individuo del grafo origen al tamaño del individuo del grafo generado. Este algoritmo parte de los dos grafos de origen distintos de los cuales se selecciona uno dependiendo del tipo de grupo para el que se están generando las conexiones.

En la segunda etapa se generan las conexiones entre los diferentes grupos utilizando un algoritmo de generación de conexiones aleatorio. Para generar estas conexiones se calcula en número de conexiones que deben generarse entre dos colectivos y se generan aleatoriamente. A continuación se muestra el pseudocódigo del algoritmo generado de conexiones entre grupos:

```
Desde colectivoA = 0 hasta colectivo = total_colectivos
  Desde colectivoB = 0 hasta colectivo = total_colectivos
    //Nota: si (colectivoA != colectivoB)
    Si ( probabilidad < prob_grupo_fuerte ) {
      Obtener grafo de tamaño mínimo (colectivoA, colectivoB)
      Obtener porcentaje de conexiones entre tipo_colectivoA y
      tipoColectivoB para conexión tipo fuerte
      Número de conexiones = porcentaje de conexiones * tamaño
      de grafo mínimo
      Generar número de conexiones al azar entre individuos de
      ambos colectivos
    }
  Fin Si
Fin Desde
Fin Desde
```

El sistema distingue dos tipos de conexiones entre grupos conexiones fuertes y conexiones débiles. Estos tipos de conexiones representa el nivel de relación entre dos colectivos estando más ligados aquellos cuyo tipo de conexiones sean fuertes. Una vez determinado mediante la probabilidad de que dos colectivos tengan un tipo de conexión u otro se obtiene el número de conexiones que deben existir entre ambos colectivos y se generan aleatoriamente el número de conexiones establecido por ese valor.

El objetivo de esta parte del trabajo es implementar un algoritmo cuyo grafo resultante mantenga las propiedades del grafo de origen y cuyas conexiones sean más realistas que las del grafo generado por el algoritmo de escalado aleatorio comentado anteriormente.

Para el desarrollo de esta parte del trabajo, en primer lugar, se ha realizado un estudio bibliográfico basándose en diversos documentos previos de otros autores con el fin de determinar el algoritmo a implementar y posteriormente analizar y comprar los resultados del algoritmo implementado con el algoritmo inicial.

Los diferentes algoritmos que se han estudiado son los siguientes:

- Random Walk [10]: En este algoritmo se selecciona un vértice aleatorio del grafo de origen como semilla del grafo de la muestra. A continuación se selecciona otro vértice aleatorio y se calcula la probabilidad de transición a ese vértice en caso de que sean vecinos. Una vez calculada la probabilidad se genera un número aleatorio mediante una distribución uniforme entre 0 y 1 y se compara con la probabilidad de transición. Si supera la probabilidad calculada se transita al nuevo vértice y se incrementa el coste total de la muestra por el coste de salto. En caso contrario se mantiene en el mismo vértice. Este proceso se repite hasta cumplir el coste que se haya establecido para el grafo de la muestra.
- Random Jump: Este algoritmo es muy similar al algoritmo Random Walk. La diferencia es que en este algoritmo se puede transitar a otros vértices los cuales no sean vecinos mediante una probabilidad fija.
- Albatross [1]: Este algoritmo selecciona de inicio un vértice aleatorio. En primer lugar se genera un número aleatorio mediante una distribución uniforme y se comprueba si realiza un salto aleatorio si supera una probabilidad p fija. Si se realiza el salto se

aumenta el coste de salto sobre el coste total establecido. Si no realiza el salto aleatorio se selecciona una arista que conecta el vértice actual con un vértice vecino. A continuación se calcula la probabilidad de transición entre el vértice u y su vértice vecino v . Si se supera dicha probabilidad se realiza la transición a este nuevo vértice y se aumenta el coste de transición al coste total. Si no se supera dicha probabilidad se mantiene en el vértice actual y se repite el proceso. Este proceso se repite hasta alcanzar el coste total establecido para el grafo de la muestra.

Basado en el análisis realizado en el artículo *“Albatross Sampling: Robust and Effective Hybrid Vertex Sampling for Social Graphs”* [1] se tomó la decisión de implementar el algoritmo Albatross por mantener unas propiedades muy similares a las del grafo de origen. En el artículo mencionado realizan el análisis de los diferentes algoritmos concluyendo que albatross tiene una mayor robustez al en la distribución en el grado de conexión de los vértices y una rápida convergencia con un mismo coste del grafo generado.

Una vez tomada esta decisión, se realizó la implementación del algoritmo con las variaciones necesarias para las diferentes etapas de la generación del grafo.

En la primera de las etapas, en la que se generan las conexiones dentro de un grupo, se mantiene el funcionamiento normal del algoritmo. Se crea una lista de vértices de la muestra a la que se van añadiendo iterativamente los vértices escogidos del grafo de origen. Una vez esta lista está llena, es decir, se ha alcanzado el tamaño de la población deseada, se añaden las aristas al conjunto del grafo.

Para añadir las aristas se busca en el grafo de origen las conexiones existentes entre cada uno de los vértices de la muestra y se añaden a las estructuras de almacenamiento de enlaces. Estas estructuras son dos listas en las que una se encarga de almacenar los vértices de origen de los enlaces y la otra los vértices de destino. De esta manera, para poder conocer un enlace, basta con acceder a la posición i de cada una de las listas y se obtiene el vértice de origen y destino de la arista i .

Para la generación de las conexiones entre grupos se ha modificado el algoritmo añadiendo dos listas de vértices, una para cada grupo, debido a que los grupos no necesariamente deben tener el mismo tamaño. Una vez hecho esto, se seleccionan los vértices mediante el algoritmo Albatross añadiendo cada vértice seleccionado a ambas listas de vértices hasta completar el menor de los grupos. Una vez completado el grupo de menor tamaño se siguen seleccionando vértices pero se añaden únicamente a la lista del grupo mayor.

Una vez finalizada la selección, se añaden a la lista los vértices del grafo de origen que conectan un vértice de la lista del grupo A con los vértices de la lista del grupo B.

A continuación se muestra el pseudocódigo de cada una de las dos etapas de generación del grafo:

Primera etapa: Muestreo de conexiones intragrupalas. Este algoritmo implementa el algoritmo Albatross para cada colectivo del grafo de la muestra. Para ello se obtiene el tamaño del colectivo y se asigna como coste total del grafo a generar. Una vez echo esto se aplica el sistema explicado anteriormente para la selección de individuos hasta completar el grafo. Una vez se ha completado el tamaño establecido se almacenan las conexiones existentes entre los

individuos seleccionados del grafo de origen construyendo así las conexiones del grafo de la muestra.

```
Total-cost <- GroupSize
cost<-0
P <- 0.2
Lista de vértices de la muestra S <- Empty
Seleccionar un vértice aleatorio v del grafo de origen
While cost < Total-Cost do
  Generar el valor alpha de una distribución uniforme U[0,1]
  If alpha < P then
    Seleccionar un nuevo vértice aleatorio u del grafo de
    origen
    if u no ha sido visitado (u no se encuentra en S) then
      Añadir u a la lista de vértices S
      cost <- cost+1
    end if
    v<-u
  else
    seleccionar un nuevo vértice aleatorio w conectado con
    v
    generar beta de una distribución uniforme
    kw <- conectividad del vértice w
    kv <- conectividad del vértice v
    if beta< kw/kv then
      if w no ha sido visitado (u no se encuentra en
      S) then
        Añadir w a la lista de vértices S
        cost <- cost+1
      end if
      v<-w
    else
      mantenerse en v
      if v no ha sido visitado (v no se encuentra en
      S) then
        Añadir v a la lista de vértices S
        cost <- cost+1
      end if
    end if
  end if
end while
for i<0 to vértices en S
  seleccionar vértice i
  aristas <- número aristas del vértice i
  for t<-0 to aristas del vértice i
    destinoT <- seleccionar destino de arista t
    for k<-0 to vértices en S
      if destinoT == vértice k then
        añadir arista al conjunto de aristas de la
        muestra
      end if
    end for
  end for
end for
```

Segunda etapa: Muestreo de conexiones externas. Una vez se ha aplicado el algoritmo anterior se aplica el siguiente algoritmo para cada par de colectivos del grafo de la muestra. En primer lugar se asignan los costes totales de cada grupo como el número de individuos de cada colectivo. Posteriormente este algoritmo selecciona los individuos del grafo de la muestra siguiendo el sistema explicado anteriormente. Una vez se seleccionan todos los individuos se

almacenan las conexiones existentes entre los individuos seleccionados del grafo de origen, de tal manera que se almacenan las conexiones de cada individuo seleccionado para el grupo A con los individuos seleccionados para el grupo B.

```

Total-cost <- GroupSize (del grupo mayor A)
Total-costB <- GroupSize (del grupo menor B)
cost<-0
P <- 0.2
Lista de vértices de la muestra Sa <- Empty
Lista de vértices de la muestra Sb <- Empty
Seleccionar un vértice aleatorio v del grafo de origen
While cost < Total-Cost do
  Generar el valor alpha de una distribución uniforme U[0,1]
  If alpha < P then
    Seleccionar un nuevo vértice aleatorio u del grafo de
    origen
    if u no ha sido visitado (u no se encuentra en Sa)
    then
      Añadir u a la lista de vértices Sa
      cost <- cost+1
      if cost < Total-CostB then
        Añadir u a la lista de vértices Sb
      end if
    end if
    v<-u
  else
    seleccionar un nuevo vértice aleatorio w conectado con
    v
    generar beta de una distribución uniforme
    kw <- conectividad del vértice w
    kv <- conectividad del vértice v
    if beta< kw/kv then
      if w no ha sido visitado (u no se encuentra en
      Sa) then
        Añadir w a la lista de vértices Sa
        cost <- cost+1
        if cost < Total-CostB then
          Añadir w a la lista de vértices Sb
        end if
      end if
      v<-w
    else
      mantenerse en v
      if v no ha sido visitado (v no se encuentra en
      S) then
        Añadir v a la lista de vértices S
        cost <- cost+1
        if cost < Total-CostB then
          Añadir v a la lista de vértices Sb
        end if
      end if
    end if
  end if
end while
for i<0 to vértices en Sa
  seleccionar vértice i
  aristas <- número aristas del vértice i
  for t<-0 to aristas del vértice i
    destinoT <- seleccionar destino de arista t
    for k<-0 to vértices en Sb
      if destinoT == vértice k then

```

```
                                añadir arista al conjunto de aristas de la
                                muestra
                                end if
                            end for
                        end for
                    end for
```

Una vez implementadas estas funcionalidades en el sistema permitiendo generar nuevos grafos, se ha realizado un análisis exhaustivo de los grafos generados en comparación a los grafos de los que se disponía hasta el momento. Este análisis se detalla en la sección 6.1 de este mismo documento.

4.2. Visualización.

El objetivo de este modulo es poder mostrar los resultados obtenidos de la ejecución del simulador sobre un mapa en el que se muestre el estado de la enfermedad en cada ciudad simulada y realizar una extrapolación de los puntos al resto del país. Con esto se busca poder desarrollar una herramienta que facilite analizar la propagación de la enfermedad simulada en el territorio español.

En el comienzo del desarrollo de este módulo del sistema se definieron en primer lugar las diferentes clases que compondrían el módulo en un desarrollo orientado a objetos. Cada una de estas clases implementan los métodos necesarios para cumplir los requisitos establecidos que se detalla en la sección 5 de este documento.

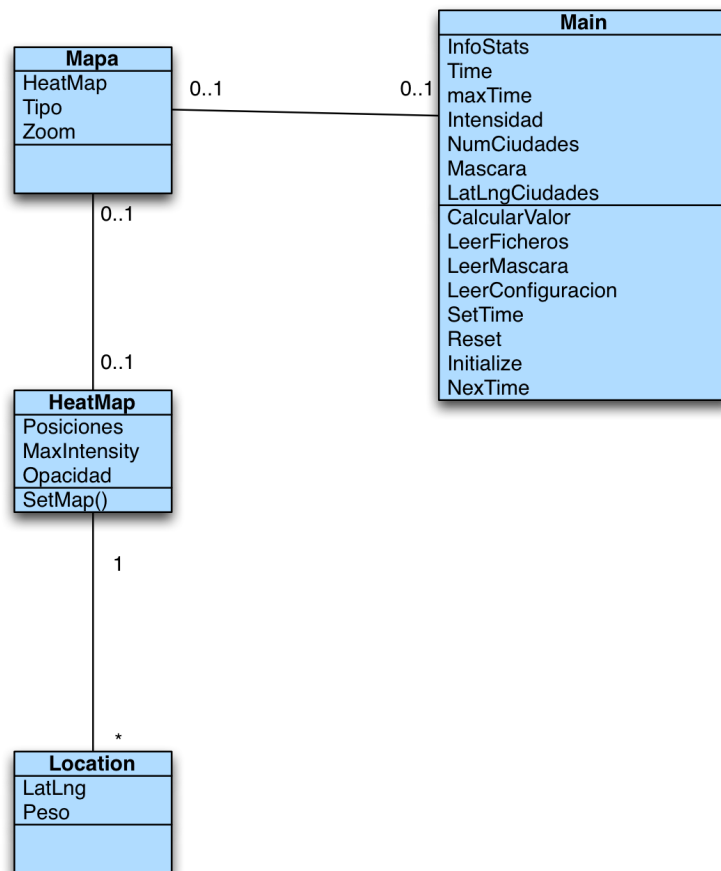


Figura 7: Diagrama de clases.

En el diagrama de la figura 7 se puede observar la estructura de clases que compone el módulo. El eje principal es la clase Main. Esta clase incorpora las funcionalidades necesarias para cargar los ficheros y calcular los valores de la matriz representada en el mapa de calor (HeatMap) para cada intervalo de tiempo. Un mapa de calor es una representación grafica de cada uno de los valores alojados en una matriz representándolos mediante una escala de colores.

Igualmente esta clase incorpora las funcionalidades de control sobre la ejecución de la visualización para reiniciar o situarla en un determinado tiempo.

Por otro lado la clase Mapa carga el recurso del mapa proporcionado por GoogleMaps que permite incorporar un mapa interactivo de la tierra en la interfaz que llevaría asociado el mapa de calor calculado por la clase Main.

Por ultimo la clase HeatMap contiene la información de la matriz de posiciones que definen el nivel de infección de cada punto del territorio español. Este valor de infección de cada punto es un valor entre 0 y 1 que define el porcentaje de infectados con respecto a la población local.

A continuación, se describen cada una de las clases y las funciones que implementan:

Clase	Main
Atributos	<ul style="list-style-type: none"> - InfoStats: Almacena la información de los datos de resultado leídos para un tiempo t. - Time: Contador que define el tiempo t actual. - maxTime: Entero que define el tiempo del final de la simulación. - Intensidad: Define la intensidad máxima del mapa de calor. - NumCiudades: Almacena el número de ciudades de la simulación. - Mascara: Almacena en un vector las posiciones que pertenecen a tierra y mar de la matriz del mapa de calor. En este vector la tierra se almacena como 1 y el mar como 0. - LatLngCiudades: Almacena en un vector las latitudes y longitudes de las ciudades de la simulación.
Funciones	<ul style="list-style-type: none"> - CalcularValor(): Implementa la funcionalidad que calcula el valor del peso de cada punto del mapa de calor. - LeerFicheros(): Función que implementa la lectura de los ficheros de resultado de la simulación realizada. - LeerMascara(): Función que lee el archivo que almacena la máscara y la carga en el atributo correspondiente. - LeerConfiguracion(): Implementa la lectura de los ficheros de configuración como son el número de ciudades y las ubicaciones. - SetTime(): Función que modifica el tiempo actual a un tiempo introducido por el usuario. - ResetTime(): Función que reinicializa el tiempo al primer instante de la simulación. - Initialize(): Función que realiza la precarga del mapa y deja dispuesto todo lo necesario para comenzar la visualización. - NexTime(): Función que realiza los cambios necesarios en el mapa de calor y actualiza el mapa para mostrar un nuevo tiempo.

Tabla 2: Clase main.

Clase	Mapa
Atributos	<ul style="list-style-type: none"> - HeatMap: Atributo que almacena el mapa de calor mostrado en el mapa. - Tipo: Atributo que almacena el tipo de mapa a mostrar, ya sea satélite o de carretera. - Zoom: Atributo que almacena el zoom al que se encuentra el mapa.
Funciones	

Tabla 3: Clase mapa.

Clase	HeatMap
-------	---------

Atributos	<ul style="list-style-type: none"> - Posiciones: Atributo que almacena una matriz de "posición". - MaxIntensity: Intensidad máxima que se contempla en este mapa de calor. - Opacidad: Marca la opacidad que muestra esta capa sobre el mapa.
Funciones	<ul style="list-style-type: none"> - SetMap(): Función que muestra la capa del mapa de calor sobre el mapa.

Tabla 4: Clase heatMap

Clase	Posición
Atributos	<ul style="list-style-type: none"> - LatLng: Latitud y longitud del punto específico del mapa de calor. - Peso: valor de la intensidad local del punto.
Funciones	

Tabla 5: Clase posición

Para el desarrollo de esta aplicación se hace necesario una función de interpolación que permita calcular los valores de los puntos. La salida del simulador da en formato de archivo de texto el estado de la enfermedad en las ciudades simuladas. Por ello se hace necesario encontrar una función que permita calcular el valor de los puntos intermedios entre las ciudades a partir de los valores que proporciona la salida del simulador para cada ciudad.

Por este motivo se evaluaron diferentes funciones para evaluar el peso correspondiente a cada punto de la matriz del mapa de calor con el fin de obtener una interpolación entre los focos de las ciudades.

Para ello se aplicaron diferentes funciones basadas en el algoritmo de clasificación k-NN. Este algoritmo mediante un proceso de aprendizaje va seleccionando de las instancias de entrenamiento aquellas que permiten clasificar al resto. El conjunto de instancias seleccionadas se denominan prototipos. En el caso particular de este proyecto se toman las ciudades como si fueran los prototipos del algoritmo clasificador.

Una vez seleccionados los prototipos este algoritmo da el valor a una nueva instancia, cuyo valor es desconocido, teniendo en cuenta los k vecinos más cercanos. En la figura 8 se muestra gráficamente el funcionamiento de este algoritmo con un valor de k=3.

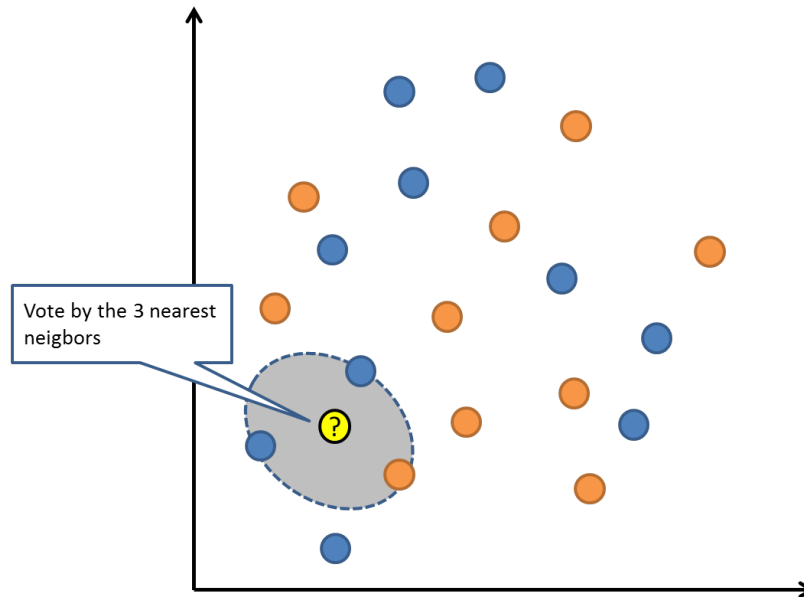


Figura 8: Representación k-NN.

En el caso de valores continuos la clasificación se realiza mediante una votación que pesa el valor de cada vecino por la distancia:

$$x = \frac{\sum_{i=1}^k w_i * y_i}{\sum_{i=1}^k w_i}$$

$$w_i = \frac{1}{d(x, y_i)^2}$$

donde x es el valor del punto a clasificar,

y_i el valor del vecino i más cercano y w_i el peso de la votación del vecino i

En el caso particular de esta aplicación los valores que tomaran tanto y como x serán valores entre 0 y 1 que representan el porcentaje de infección del punto con respecto a al máximo de infección total. Partiendo de esta función se tomaron como k los 3 vecinos más cercanos y se realizaron las siguientes variantes sobre el 'voto' que aporta cada vecino para un nuevo punto:

- $1/\text{Distancia}^2$: Esta es la función de peso básica de k-NN (w_i). Tras implementar esta función se comprobó que el efecto que tiene un foco sobre los diferentes puntos del mapa es demasiado elevado.

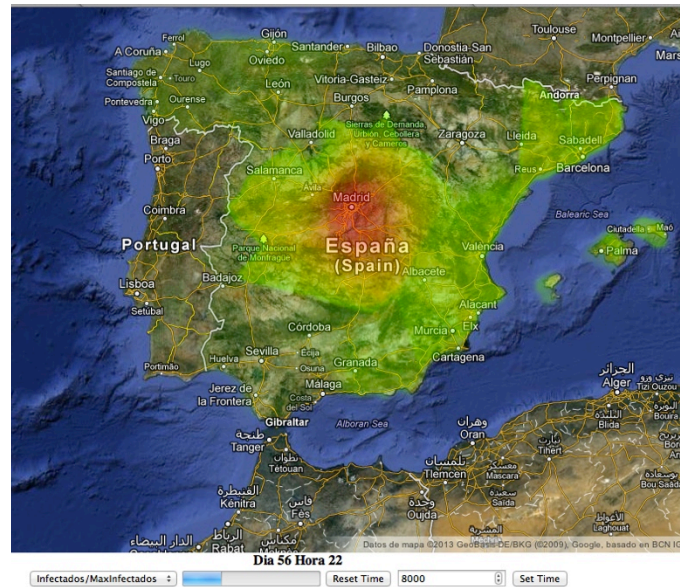


Figura 9: Visualización $1/\text{distancia}^2$

En la figura 9 se puede observar como el valor de Madrid abarca una gran área sin variar prácticamente el valor del foco y minimizando el efecto sobre un punto el efecto de otros focos como en el caso de Valladolid cuya área, al tener un porcentaje de infección igual a 0, es mínimo.

- $1/\text{Distancia}^3$: Con esta función se busca disminuir el área de efecto de un foco con el fin de obtener una visualización más precisa de lo que está ocurriendo.



Figura 10: Visualización $1/\text{distancia}^3$

Como se puede observar en la figura 10 queda mejor interpolado los valores del área de Madrid, sin embargo, fijándose en el área de Barcelona puede observarse que tiene la misma

importancia la capital de provincia que sus ciudades vecinas como es Hospitalet. Esto implica que el foco de Barcelona no tenga un área de efecto significativo cuando su población es muy superior a las ciudades vecinas.

En este punto tanto Barcelona como Madrid alcanzan el máximo punto de la enfermedad (rojo intenso) . Sin embargo Barcelona se muestra con un nivel medio de infección debido a la influencia de las ciudades vecinas. Esto implica que esta función para la clasificación sea inaceptable.

- Población/Distancia³: Con motivo de solucionar el problema anterior se pesó la interpolación en función de la población y la distancia de las ciudades vecinas. De esta manera se obtiene una visualización más precisa. Sin embargo, en este caso las diferencias de poblaciones son muy grandes y la mayoría de focos pierden importancia frente a las grandes ciudades. En la figura 11 se puede observar como el valor de infección de Madrid afecta a un área excesivamente grande al perder importancia las ciudades vecinas por la gran diferencia de población. También puede observarse como se soluciono el problema existente en Barcelona que con esta modificación muestra correctamente el valor de la zona.

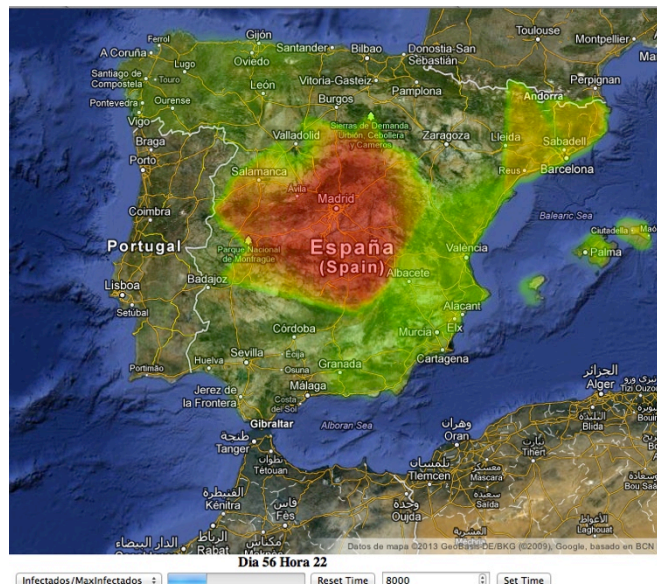


Figura 11: Visualización población/distancia³

- Población^{1/2}/Distancia³: Con el fin de subsanar el anterior problema se definió la función que pesa al igual que en el caso anterior por la población, pero, con el fin de disminuir las diferencias entre grandes y pequeñas poblaciones, se introdujo el valor de la población como la raíz cuadrada. Con esta función se consiguió una visualización bastante precisa y una interpolación que permite comprender como se extiende la enfermedad por las distintas ciudades. En la figura 12 se puede observar como con esta función el área que abarca Madrid es inferior al área sobre la que afecta con la función anterior.

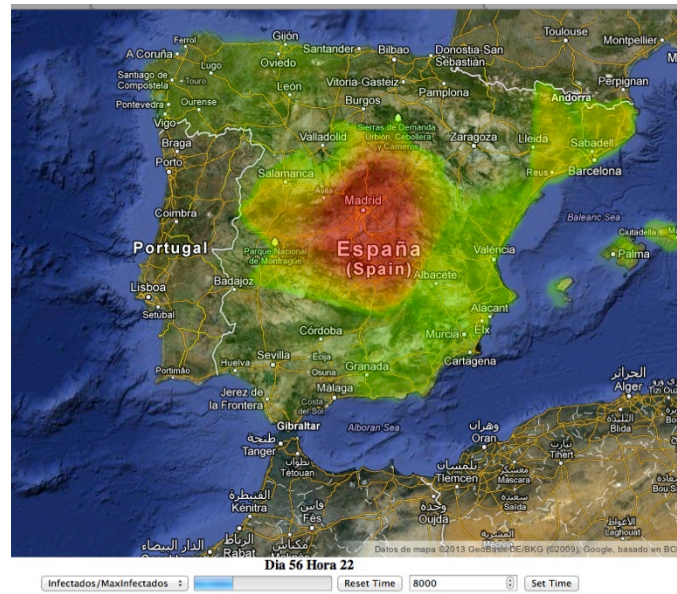


Figura 12: Visualización raíz de población/distancia³

5. Requisitos.

En esta sección se detallan los requisitos obtenidos mediante el análisis del problema para el desarrollo de la aplicación de visualización de los resultados de

5.1. Requisitos de usuario.

En esta sección se detallan los requisitos de usuario divididos en dos grupos: requisitos de capacidad y de restricción.

Los requisitos de capacidad son aquellos que determinan las funcionalidades que requieren los usuarios de la aplicación. Los requisitos de restricción son aquellos que determinan las diferentes restricciones que imponen los usuarios sobre el comportamiento de la aplicación.

Para detallar los requisitos se sigue la siguiente plantilla:

Identificador:	RQ-US-XXX-YY
Descripción:	
Necesidad:	
Prioridad:	

Tabla 6: Plantilla requisitos de usuario.

- **Identificador:** código identificador del requisito que sigue el siguiente formato RQ-US-XXX-YY. Donde YY identifica el tipo de requisito, CA para capacidad y RE para restricción, y XXX identifica el número de requisito.
- **Descripción:** detalla la especificación del requisito.
- **Necesidad:** establece la importancia del cumplimiento del requisito para el cliente. Podrá tomar los valores alta, media o baja.
- **Prioridad:** establece la importancia del requisito dentro del sistema desde el punto de vista del desarrollo. El valor de este campo podrá ser alta, media o baja

5.1.1. Requisitos de usuario de capacidad.

A continuación se muestran los requisitos de usuario de capacidad obtenidos tras el análisis de requisitos realizado.

Identificador:	RQ-001-CA
Descripción:	Deberán mostrarse sobre un mapa los resultados de la ejecución del simulador.
Necesidad:	Alta
Prioridad:	Alta

Tabla 7: RQ-001-CA.

Identificador:	RQ-002-CA
Descripción:	Deberá poder detenerse la visualización en un punto de la ejecución.

Necesidad:	Media
Prioridad:	Media

Tabla 8: RQ-002-CA.

Identificador:	RQ-003-CA
Descripción:	Deberá poder reanudarse la ejecución
Necesidad:	Media
Prioridad:	Media

Tabla 9: RQ-003-CA.

Identificador:	RQ-004-CA
Descripción:	Deberá poderse mostrar la visualización de diferentes parámetros de la simulación.
Necesidad:	Alta
Prioridad:	Media

Tabla 10: RQ-004-CA.

Identificador:	RQ-005-CA
Descripción:	Deberá poder reiniciarse la simulación desde el punto inicial.
Necesidad:	Media
Prioridad:	Baja

Tabla 11: RQ-005-CA.

Identificador:	RQ-006-CA
Descripción:	La visualización permitirá hacer zoom en tiempo de ejecución
Necesidad:	Baja
Prioridad:	Baja

Tabla 12: RQ-006-CA.

Identificador:	RQ-007-CA
Descripción:	El usuario deberá poder saber en que momento de la ejecución se encuentra.
Necesidad:	Media
Prioridad:	Baja

Tabla 13: RQ-007-CA.

Identificador:	RQ-008-CA
Descripción:	Deberá mostrarse los valores importantes sobre la ejecución como son el número de infectados, fallecidos y hospitalizados.
Necesidad:	Media
Prioridad:	Media

Tabla 14: RQ-008-CA.

Identificador:	RQ-009-CA
Descripción:	El usuario deberá poder cambiar los datos de una

	simulación por los de otra de forma automatizada.
Necesidad:	Alta
Prioridad:	Media

Tabla 15: RQ-009-CA.

5.1.2. Requisitos de usuario de restricción.

En esta sección se muestran los requisitos de usuario de restricción establecidos para la aplicación.

Identificador:	RQ-001-RE
Descripción:	El tiempo de inicio de la aplicación no deberá exceder los 5 segundos.
Necesidad:	Media
Prioridad:	Media

Tabla 16: RQ-001-RE.

Identificador:	RQ-002-RE
Descripción:	El tiempo de cambio de tiempo en la visualización no deberá exceder los 4 segundos.
Necesidad:	Media
Prioridad:	Media

Tabla 17: RQ-002-RE.

Identificador:	RQ-003-RE
Descripción:	La aplicación deberá poder ejecutarse sobre cualquier sistema operativo.
Necesidad:	Alta
Prioridad:	Alta

Tabla 18: RQ-003-RE.

5.2. Casos de uso.

En esta sección se describen los casos de uso extraídos de los requisitos de usuario funcionales. El objetivo de los casos de usos es detallar gráficamente las funcionalidades y tareas que el usuario debe poder realizar sobre el sistema. Junto al detallado gráfico de los casos de uso se añade una tabla que describe los pasos que a realizar para completar cada una de las tareas.

El siguiente diagrama de la figura 13 se muestran los diferentes casos de uso que implementara el sistema y que definen la interacción del usuario con el sistema. En este diagrama se muestra el usuario relacionado con los diferentes casos de uso dentro del sistema que puede ejecutar mediante una línea. Dentro del propio sistema los casos de uso que dependen de la ejecución de otro aparecen relacionados mediante una flecha negra identificada con la etiqueta “<extiende>”.

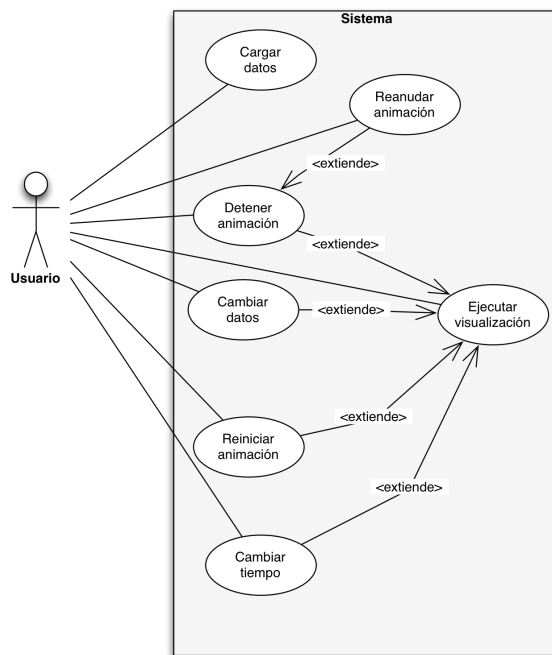


Figura 13: Diagrama de casos de uso.

A continuación se describen los casos de uso que se detallarán siguiendo la plantilla mostrada a continuación:

Nombre	Identificador	CU-XXX
Descripción		
Actor		
Precondiciones		
Postcondiciones		
Flujo normal		
Flujo alternativo		

Tabla 19: Plantilla casos de uso.

- **Nombre:** nombre del caso de uso que se describe.
- **Identificador:** Código identificador del caso de uso que sigue el formato CU-XXX. Donde CU identifica que es un caso de uso y XXX es el número identificativo.
- **Actor:** Identifica al actor que lleva a cabo el caso de uso. Puesto que se trata con sólo un tipo de usuario el actor para todos los casos de uso será Usuario.
- **Precondiciones:** Condiciones necesarias e iniciales para poder realizar la secuencia de acciones.

- **Postcondiciones:** Condiciones que se obtienen tras haber realizado la secuencia de acciones.
- **Flujo normal:** Secuencia de acciones a realizar.
- **Flujo alternativo:** Secuencia, alternativa al flujo normal, de acciones a poder realizar. En caso de no existir ningún flujo alternativo se indicara que no aplica.

Nombre	Cargar datos	Identificador	CU-001
Descripción	El usuario realiza una carga de los datos de una simulación en la aplicación del visualizador.		
Actor	Usuario		
Precondiciones	El usuario debe tener los datos de salida de una ejecución del simulador.		
Postcondiciones	El sistema tiene cargados los datos de la simulación		
Flujo normal	<ol style="list-style-type: none"> 1. Procesar los datos al formato establecido para el visualizador. 2. Introducir los datos dentro de la carpeta de carga. 3. Ejecutar la función de carga del sistema. 4. Modificar la configuración al número de ciudades simuladas. 		
Flujo alternativo	No aplica.		

Tabla 20: CU-001.

Nombre	Ejecutar visualización	Identificador	CU-002
Descripción	El usuario ejecuta la visualización de una simulación.		
Actor	Usuario		
Precondiciones	El usuario debe haber realizado la carga de los ficheros de salida del simulador en el sistema.		
Postcondiciones	El usuario visualiza los resultados de la simulación en el mapa.		
Flujo normal	<ol style="list-style-type: none"> 1. El usuario inicia la aplicación. 2. El sistema muestra el mapa e inicia la animación mostrando el mapa de calor. 3. El sistema muestra el resumen de datos del tiempo actual. 		
Flujo alternativo	No aplica.		

Tabla 21: CU-002.

Nombre	Detener animación	Identificador	CU-003
Descripción	El usuario detiene la animación en un punto.		
Actor	Usuario		

Precondiciones	El usuario debe haber iniciado una visualización (CU-002).
Postcondiciones	El usuario visualiza el estado de la epidemia en el tiempo que ha detenido la ejecución.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario selecciona la opción “congelado”. 2. El sistema congela la animación en el punto actual.
Flujo alternativo	No aplica.

Tabla 22: CU-003.

Nombre	Reiniciar animación	Identificador	CU-004
Descripción	El usuario busca reiniciar la animación desde el punto inicial.		
Actor	Usuario		
Precondiciones	El usuario debe haber iniciado una visualización (CU-002).		
Postcondiciones	El sistema vuelve al tiempo inicial de la simulación y reinicia la animación desde ese punto.		
Flujo normal	<ol style="list-style-type: none"> 1. El usuario selecciona la opción “reiniciar”. 2. El sistema vuelve al estado en el tiempo 0 y vuelve a ejecutar la animación. 		
Flujo alternativo	No aplica.		

Tabla 23: CU-004.

Nombre	Cambiar tiempo	Identificador	CU-005
Descripción	El usuario quiere pasar de un instante de tiempo de la simulación a otro.		
Actor	Usuario		
Precondiciones	El usuario debe haber iniciado una visualización (CU-002).		
Postcondiciones	El sistema pasa al tiempo introducido y continúa la animación desde ese punto.		
Flujo normal	<ol style="list-style-type: none"> 1. El usuario introduce el valor de un instante de tiempo y selección la opción de “seleccionar tiempo”. 2. El sistema comprueba el valor introducido. 3. Si es correcto el sistema cambia el instante de la simulación en que se encuentra por el instante introducido. 		
Flujo alternativo	<ol style="list-style-type: none"> 3.a. Si el valor introducido es incorrecto el sistema muestra un mensaje indicando el error y los valores permitidos. 3.b. El usuario cierra el mensaje y vuelve a 		

introducir el valor (Paso 1 del flujo normal).

Tabla 24: CU-005.

Nombre	Cambiar datos	Identificador	CU-006
Descripción	El usuario cambia los datos de la simulación que se muestran en el mapa.		
Actor	Usuario		
Precondiciones	El usuario debe haber iniciado una visualización (CU-002).		
Postcondiciones	El sistema muestra los datos deseados de la simulación.		
Flujo normal	<ol style="list-style-type: none"> 1. El usuario selecciona la opción de los datos que desea visualizar entre las opciones disponibles. 2. El sistema continúa la simulación desde el punto actual mostrando los datos seleccionados. 		
Flujo alternativo	No aplica.		

Tabla 25: CU-006.

Nombre	Zoom	Identificador	CU-007
Descripción	El usuario realiza un zoom sobre el mapa ampliando o alejando una zona del mismo.		
Actor	Usuario		
Precondiciones	El usuario debe haber iniciado una visualización (CU-002).		
Postcondiciones	El sistema muestra la sección del mapa ampliada.		
Flujo normal	<ol style="list-style-type: none"> 1. El usuario selecciona la opción “Zoom In” o “Zoom Out” del mapa. 2. El sistema calcula el valor umbral a partir del valor de zoom del mapa. El valor umbral es inversamente proporcional al valor de zoom. 3. El sistema recalcula los valores del mapa de calor. 4. El sistema muestra el mapa de calor actualizado. 5. El sistema muestra el resumen de datos del tiempo actual. 		
Flujo alternativo	No aplica.		

Tabla 26: CU-007.

Nombre	Reanudar animación	Identificador	CU-008
Descripción	El usuario detiene la animación en un punto.		
Actor	Usuario		
Precondiciones	El usuario debe haber iniciado una visualización		

	(CU-002) y haberla detenido previamente (CU-003).
Postcondiciones	El sistema continua la ejecución desde el punto en que se encontraba congelado.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario deselecciona la opción "congelado". 2. El sistema continúa la animación desde el punto en que se congeló.
Flujo alternativo	No aplica.

Tabla 27: CU-008.

5.3. Requisitos software.

En esta sección se describe la especificación de los requisitos software resultantes del análisis de las funcionalidades y restricciones descritas en los requisitos de usuario.

Los requisitos software se dividen en dos grupos:

- **Requisitos software funcionales:** especifican la funcionalidad del sistema.
- **Requisitos software no funcionales:** complementan los requisitos funcionales especificando las restricciones sobre el sistema y los aspectos visibles por el usuario.

Los requisitos software se especificaran siguiendo la siguiente plantilla:

Identificador:	RQ-SW-XXX-YY
Descripción:	
Necesidad:	
Prioridad:	

Tabla 28: Plantilla requisitos software.

- **Identificador:** código identificativo del requisito que sigue el formato RQ-SW-XXX-YY. Donde XXX es el número identificativo e YY indica el tipo siendo FU para los requisitos funcionales y NF para los requisitos no funcionales.
- **Descripción:** describe el requisito de una forma clara y concisa.
- **Necesidad:** indica la importancia del requisito dentro del sistema. Los valores que puede tomar son alta, media y baja.
- **Prioridad:** indica la importancia del requisito dentro del desarrollo del sistema. Los valores que puede tomar son alta, media y baja.

5.3.1. Requisitos software funcionales.

A continuación se enuncian los requisitos de software que definen la funcionalidad del sistema.

Identificador:	RQ-SW-001-FU
Descripción:	El sistema realizará la carga de los datos seleccionados en un menú desplegable leyendo la columna solicitada de los ficheros.
Necesidad:	Alta.
Prioridad:	Media.

Tabla 29: RQ-SW-001-FU.

Identificador:	RQ-SW-002-FU
Descripción:	El sistema reiniciará el contador de tiempo al solicita el “reset” de la animación.
Necesidad:	Media.
Prioridad:	Baja.

Tabla 30: RQ-SW-002-FU.

Identificador:	RQ-SW-003-FU
Descripción:	El sistema detendrá el contador de tiempo tras ejecutar la función “congelar” y no avanzará en la simulación.
Necesidad:	Media.
Prioridad:	Baja.

Tabla 31: RQ-SW-003-FU.

Identificador:	RQ-SW-004-FU
Descripción:	El sistema ejecutara cada intervalo de 3 segundos la función de interpolación y mostrara el estado en el tiempo determinado por el contador.
Necesidad:	Alta.
Prioridad:	Alta.

Tabla 32: RQ-SW-004-FU.

Identificador:	RQ-SW-005-FU
Descripción:	El sistema continuara el incremento después de detenerlo al deselectionar la opción de “congelado”.
Necesidad:	Media.
Prioridad:	Media.

Tabla 33: RQ-SW-005-FU.

Identificador:	RQ-SW-006-FU
Descripción:	Al realizar la función de zoom sobre el mapa el sistema calculara los umbrales para mantener la intensidad.
Necesidad:	Baja.
Prioridad:	Baja.

Tabla 34: RQ-SW-006-FU.

Identificador:	RQ-SW-007-FU
Descripción:	El sistema mostrará el estado de la simulación en un breve resumen con el número de infectados, hospitalizados y fallecidos en el tiempo mostrado.
Necesidad:	Alta.
Prioridad:	Alta.

Tabla 36: RQ-SW-007-FU.

Identificador:	RQ-SW-008-FU
Descripción:	El sistema realizará la carga de los datos en la estructura de ficheros de salida del simulador y lo modificará al sistema de ficheros del visualizador.
Necesidad:	Alta.
Prioridad:	Alta.

Tabla 37: RQ-SW-008-FU.

Identificador:	RQ-SW-009-FU
Descripción:	El tiempo recogerá el número introducido en el campo de texto y modificará el contador al valor solicitado.
Necesidad:	Media.
Prioridad:	Media.

Tabla 38: RQ-SW-009-FU.

Identificador:	RQ-SW-010-FU
Descripción:	El sistema mostrará en una barra de progreso la situación de la simulación.
Necesidad:	Media.
Prioridad:	Baja.

Tabla 39: RQ-SW-010-FU.

Identificador:	RQ-SW-011-FU
Descripción:	El sistema permitirá la modificación en el número de ciudades de la simulación
Necesidad:	Alta.
Prioridad:	Alta.

Tabla 40: RQ-SW-011-FU.

5.3.2. Requisitos software no funcionales.

En esta sección se describen los requisitos de software no funcionales establecidos para el desarrollo de la aplicación.

Identificador:	RQ-SW-001-NF
Descripción:	El sistema realizará una precarga del mapa al inicio de la ejecución para minimizar el tiempo de computo.
Necesidad:	Alta.
Prioridad:	Alta.

Tabla 41: RQ-SW-001-NF.

Identificador:	RQ-SW-002-NF
Descripción:	El sistema no tardará más de 5 segundos en inicializar el mapa.
Necesidad:	Alta.
Prioridad:	Alta.

Tabla 42: RQ-SW-002-NF.

Identificador:	RQ-SW-003-NF
Descripción:	El tiempo de refresco entre intervalos no deberá superar un segundo.
Necesidad:	Alta.
Prioridad:	Media.

Tabla 43: RQ-SW-003-NF.

Identificador:	RQ-SW-004-NF
Descripción:	El lenguaje utilizado será HTML y JavaScript haciendo del sistema una aplicación multiplataforma.
Necesidad:	Alta.
Prioridad:	Alta.

Tabla 44: RQ-SW-004-NF.

Identificador:	RQ-SW-005-NF
Descripción:	El sistema trabajará sobre el sistema de ficheros local.
Necesidad:	Alta.
Prioridad:	Alta.

Tabla 45: RQ-SW-005-NF.

Identificador:	RQ-SW-006-NF
Descripción:	El mapa usará las librerías de GoogleMaps para las localizaciones.
Necesidad:	Alta.
Prioridad:	Alta.

Tabla 46: RQ-SW-006-NF.

Identificador:	RQ-SW-007-NF
Descripción:	El sistema deberá mostrar si existen errores mediante ventanas PopUp para ofrecer feedback al usuario.
Necesidad:	Media.
Prioridad:	Media.

Tabla 47: RQ-SW-007-NF.

5.4. Matriz de trazabilidad.

En esta sección se describe la matriz de trazabilidad realizada para comprobar la calidad de los casos de uso y que todos los requisitos de usuario de capacidad se encuentran implementados en al menos un caso de uso. De esta manera se podrá verificar que los casos de uso detallados implementan todas las funcionalidades requeridas por el cliente.

A continuación se muestra la tabla que representa los casos de uso frente a los requisitos de usuario de capacidad indicando en que casos de uso está implementado cada requisito.

Requisitos de usuario\Casos de uso	CU-001	CU-002	CU-003	CU-004	CU-005	CU-006	CU-007	CU-008
RQ-US-001-CA		X	X	X	X	X	X	X
RQ-US-002-CA			X					
RQ-US-003-CA				X				X
RQ-US-004-CA						X		
RQ-US-005-CA					X			
RQ-US-006-CA							X	
RQ-US-007-CA		X	X	X	X	X	X	X
RQ-US-008-CA		X	X	X	X	X	X	X
RQ-US-009-CA	X							

Tabla 48: Matriz de trazabilidad.

Después de analizar esta matriz se puede comprobar que no queda ningún requisito de usuario sin cubrir por lo que se puede asegurar la calidad de los casos de uso definidos para el sistema.

6. Evaluación.

6.1. Evaluación del algoritmo de muestreo.

En esta sección se describe el análisis de los grafos generados con el algoritmo Albatross en comparación al algoritmo de escalado implementado en la versión inicial del simulador. Para realizar esta comparación se analizará tanto la estructura resultante de las conexiones como el número de conexiones.

Igualmente se compararán los resultados de la infección al pasar el grafo por el simulador EpiGraph y la complejidad de los algoritmos en función del tamaño del grafo.

Para realizar este análisis se han realizado diversas pruebas. En primer lugar con el fin de analizar el efecto de la topología resultante por cada algoritmo en los primeros días de infección se han realizado diversas pruebas con diferentes tamaños de población de los 5 días iniciales de la simulación.

A continuación a esto se ha realizado simulaciones de 100 días con el fin de comprobar el comportamiento de la enfermedad en grafos generados por los dos algoritmos en un periodo largo de tiempo y tener una idea más clara de los efectos a largo plazo.

Para finalizar, se ha realizado una última comparación del efecto de la topología en grafos con un número similar de conexiones y un mismo tamaño de población con el fin de comprobar cual es el efecto únicamente de la topología y no del número de conexiones.

Simulación de 5 días.

Población de 250

En primer lugar se realizó esta comparación con un conjunto poblacional de un tamaño de 250 individuos con un solo grupo interno en la población.

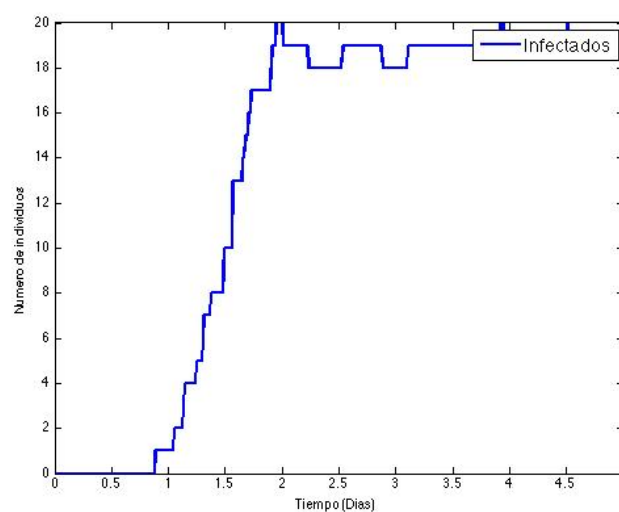


Figura 14: Epidemia con población 250 en 5 días con Escalado.

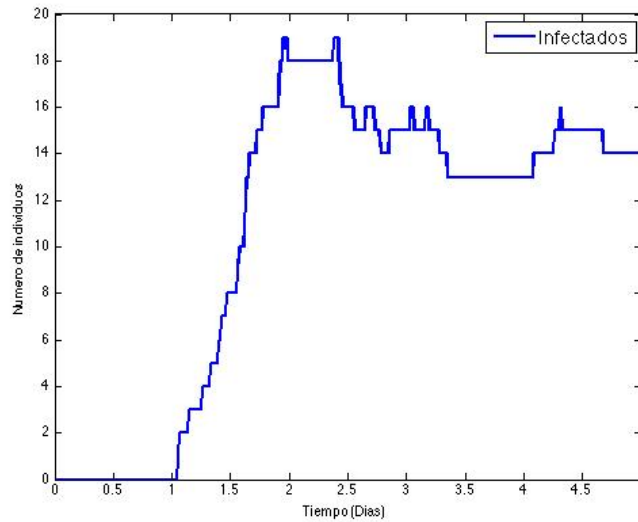


Figura 15: Epidemia con poblacion 250 en 5 días con Albatross.

En las figuras anteriores (figuras 14 y 15) se muestra el número de individuos infectados a lo largo del tiempo en los primeros 5 días de infección. Se puede observar que en el grafo generado por Albatross se da una mayor cantidad de picos de infección y no se alcanza el mismo número de individuos. Esto es debido a que en una población pequeña el número de relaciones generadas por el algoritmo Albatross es muy inferior a las conexiones generadas por el algoritmo de Escalado.

A continuación se muestra la estructura resultante de la generación de los grafos. En estas gráficas se muestran en los ejes x e y los diferentes individuos y se marca con un punto azul las relaciones existentes.

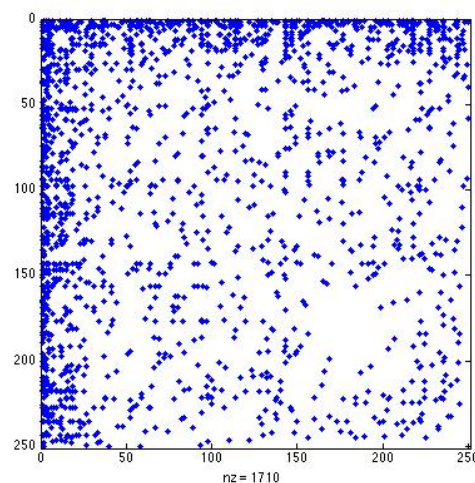


Figura 16: Matriz de adyacencia de 250 individuos mediante Escalado.

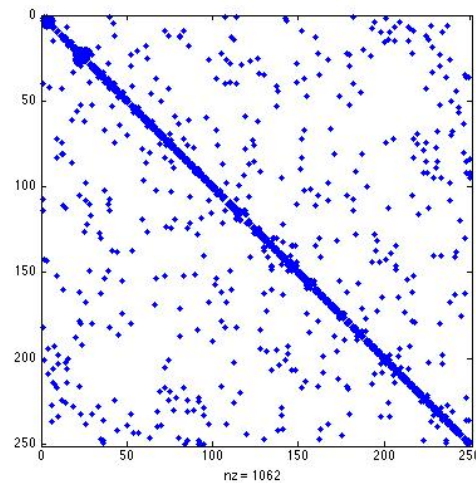


Figura 17: Matriz de adyacencia de 250 individuos mediante Albatross.

En estas gráficas se puede observar visualmente la diferencia existente en las estructuras generadas por cada algoritmo. En la gráfica generada por el algoritmo de escalado se puede observar que las conexiones se agrupan en los extremos del grupo siendo los primeros vértices los más conectados. Sin embargo, a pesar de que el número de conexiones es muy inferior, en el grafo generado por el algoritmo Albatross se puede observar como la mayoría de las conexiones se concentran en la diagonal de la matriz de conexiones del grafo.

Población de 1000

En estas nuevas gráficas se han generado grafos con 1000 individuos. En estas gráficas de infección se puede observar como, a pesar de que Albatross, para este tamaño de población, genera menos conexiones el número de infectados es mayor que en el grafo de generado por el algoritmo de escalado. Esto viene debido a la diferencia en la topología de las conexiones dentro del grafo. También se puede observar que el tiempo del primer pico de infección es prácticamente el mismo, aunque hay una diferencia considerable en el número de infectados.

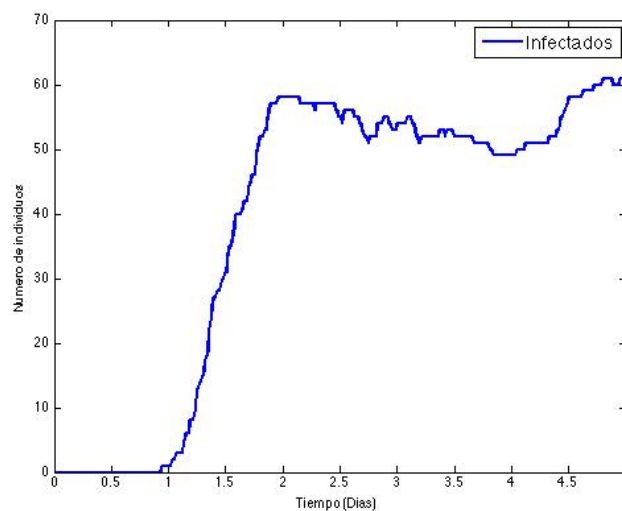


Figura 18: Epidemia con poblacion 1000 en 5 días con Escalado.

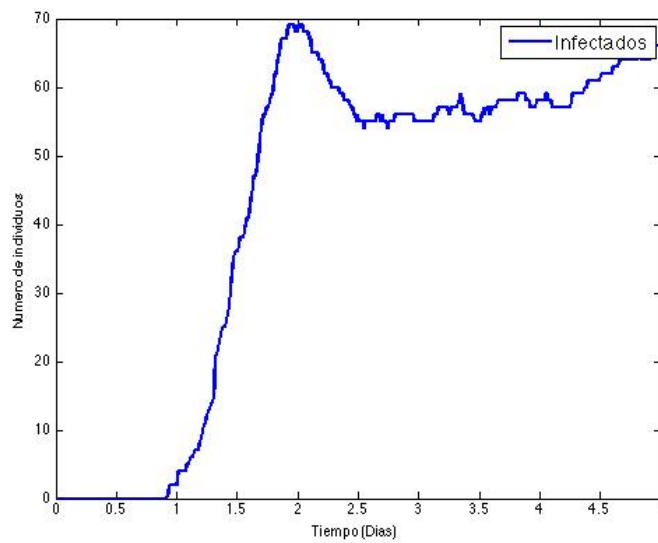


Figura 19: Epidemia con poblacion 1000 en 5 días con Albatross.

En las figuras mostradas a continuación (figuras 20 y 21) se refleja la matriz de conexiones del grafo. Estos grafos están compuestos por 3 grupos de diferentes tamaños. Se puede observar cómo la estructura que se genera con Albatross para las conexiones internas de los grupos se da también en las conexiones entre ellos. También se puede apreciar cómo, en comparación, el número de conexiones externas es superior en los grafos generados por Albatross que en los grafos generados por el algoritmo de escalado. Esto se traduce en que la enfermedad puede pasar con mayor facilidad entre un grupo y otro de individuos.

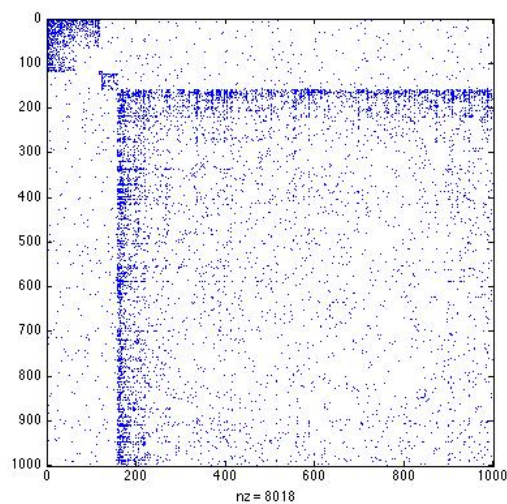


Figura 20: Matriz de adyacencia de 1000 individuos mediante Escalado.

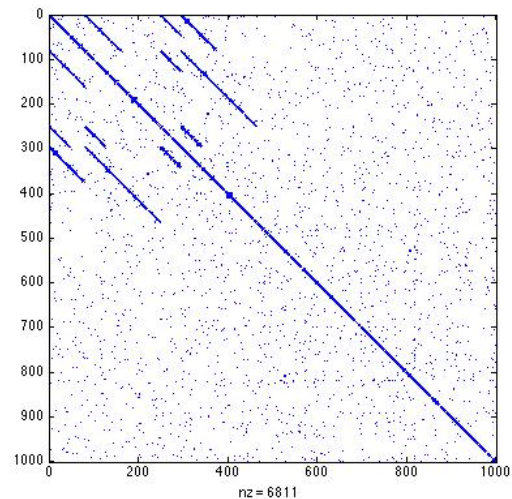


Figura 21: Matriz de adyacencia de 1000 individuos mediante Albatross.

Población 5000

En este apartado se han comparado los resultados para conjuntos poblacionales de 5000 individuos. En primer lugar el efecto que tiene en los primeros días de infección es que el primer pico de infección en el caso del grafo generado por Albatross es muy similar. En Albatross es ligeramente inferior y se produce una mayor caída que en el grafo generado por el algoritmo de escalado.

A pesar de que para este tamaño de población el número de conexiones generadas por Albatross es muy superior que las generadas por escalado, en los primeros días de la infección no repercute en una mayor infección. En las figuras 22 y 23 se muestran las graficas con los resultados de la simulación.

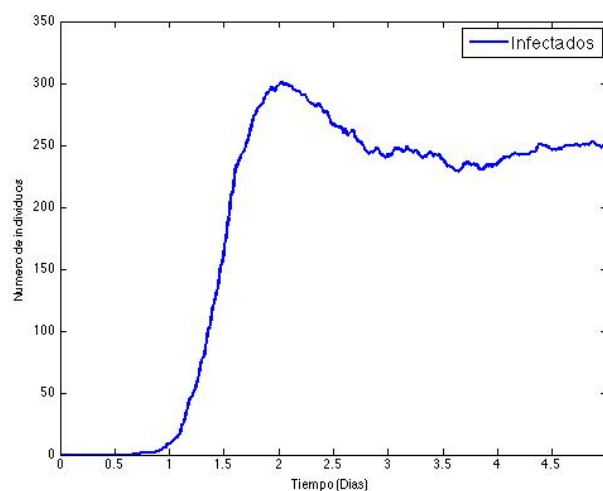


Figura 22: Epidemia con poblacion 5000 en 5 días con Escalado.

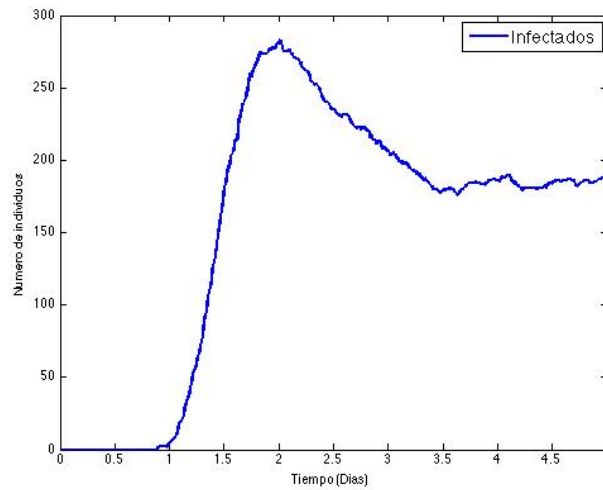


Figura 23: Epidemia con poblacion 5000 en 5 días con Albatross.

En las siguientes figuras se puede observar la topología de las conexiones generadas para este tamaño de población. En el algoritmo Albatross se observa cómo el número de conexiones ha crecido de forma exponencial, en comparación con el algoritmo de escalado, y que la estructura básica que se forma en cada grupo de conexiones del grafo se mantiene a lo largo de todas las conexiones internas y externas.

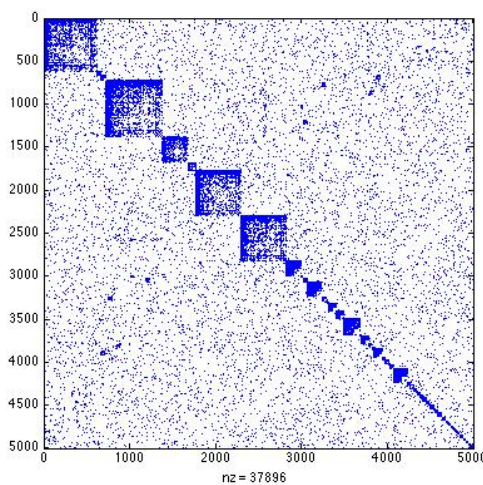


Figura 24: Matriz de adyacencia de 5000 individuos mediante Escalado.

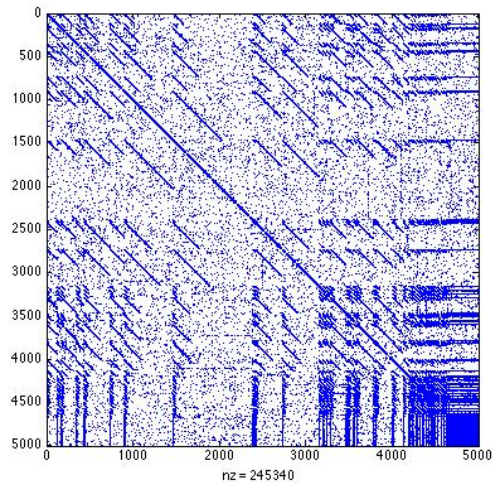


Figura 25: Matriz de adyacencia de 5000 individuos mediante Albatross.

Simulación de 100 días.

Poblacion 5000.

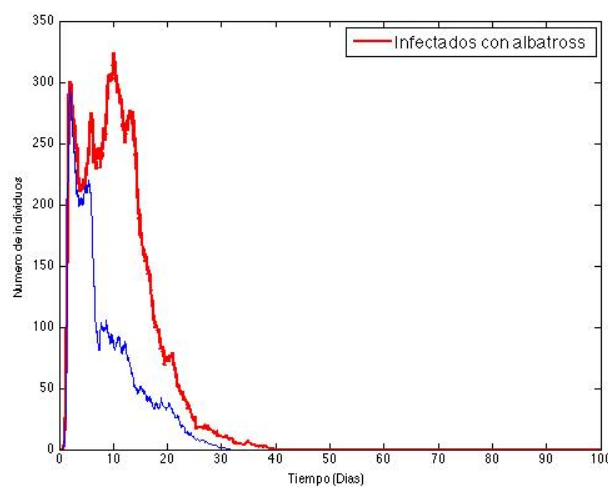


Figura 26: Epidemia 5000 individuos.

En la figura 26 se puede observar el efecto sobre la infección en el uso de un algoritmo u otro en la generación del grafo de conexiones.

Se puede observar que en grafo generado por Albatross (línea roja) la enfermedad se extiende a la misma velocidad en los cinco primeros días de infección que en el grafo generado por el algoritmo de escalado. Sin embargo, a partir de este día, el número de infectados se mantiene muy superior en el grafo generado por Albatross.

Se puede observar cómo en albatross se dan varios picos de infección que coinciden con un efecto real conocido como el efecto rebote. Esto se da también en el grafo generado por el algoritmo de escalado sin embargo es un efecto mucho más disimulado en este caso.

En la figura 27 se puede comprobar como el número de individuos que no han sufrido la enfermedad con respecto al tiempo es mucho menor con el uso de Albatross, es decir el número de infectados acumulado es mayor con Albatross. También se puede observar que en los grafos generados por albatros el tiempo que tarda en dejar de extenderse la enfermedad también es mayor en el algoritmo Albatross.

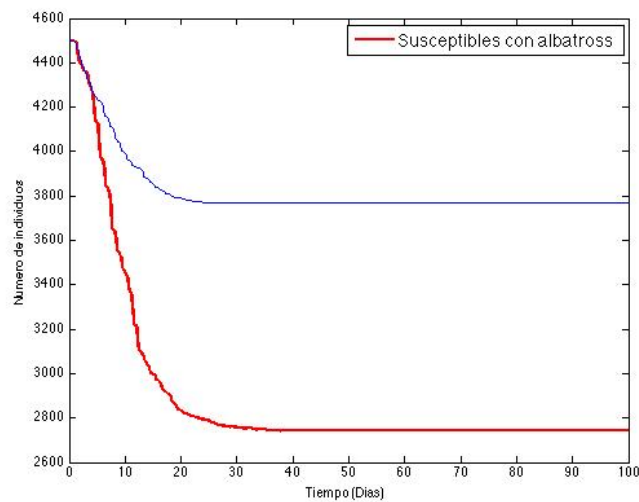


Figura 27: No infectados 5000 individuos.

Poblacion 50000.

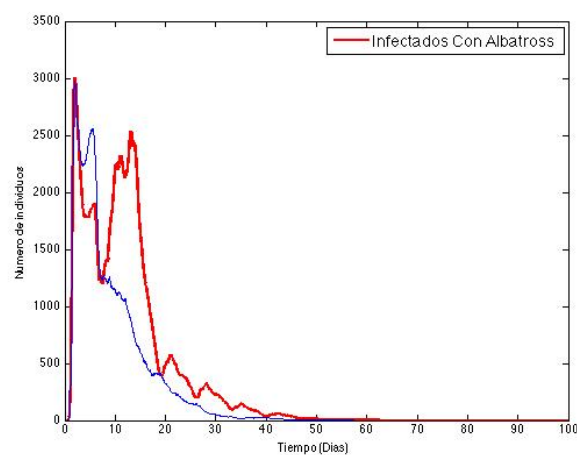


Figura 28: Epidemia 50000 individuos.

En esta gráfica se puede observar como el comportamiento de la enfermedad para el algoritmo de escalado, en comparación con la población de 5000 individuos, es el mismo a diferentes escalas. Esto quiere decir que el comportamiento a diferentes tamaños de población es lineal variando únicamente el número de individuos que se infectan pero manteniendo la forma de la función.

En el caso de Albatross sí que se da una diferencia significativa en el cambio del tamaño de la población. Se produce igualmente el efecto rebote y se infectan en conjunto un mayor

número de individuos, pero en comparación con la población de 5000 individuos se da una mayor bajada tras superar la primera ola de la infección.

Además, se puede ver que, se dan pequeños picos de infección mientras la enfermedad va cayendo en los últimos días.

Efecto de la topología.

En esta simulación, con el fin de observar el efecto que tiene únicamente la topología de los grafos de conexión de individuos, se han generado grafos con una población de 1500 individuos. Estos grafos tienen un número similar de conexiones totales lo que permite analizar el efecto únicamente de la estructura del grafo.

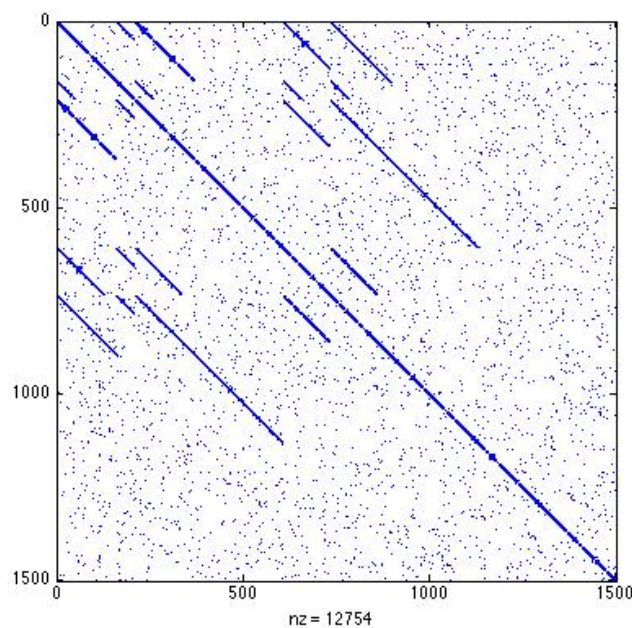


Figura 29: Topología Albatross.

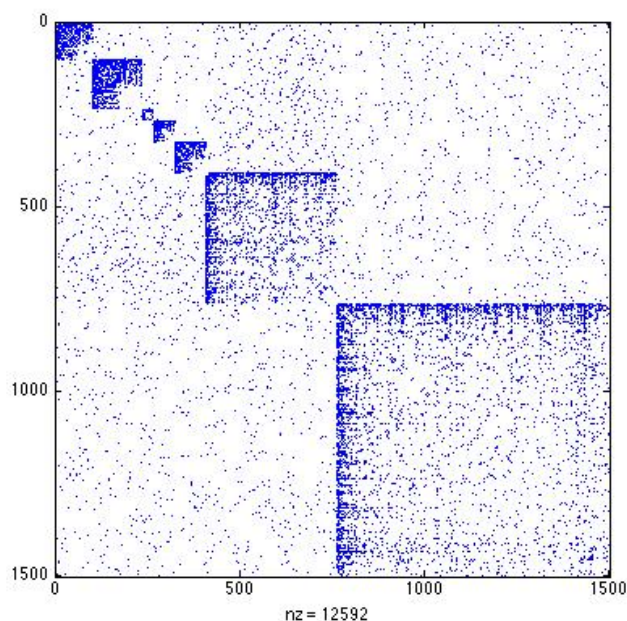


Figura 30: Topología Escalado.

En estas gráficas se puede observar visualmente la topología ya comentada anteriormente en la que se forma, utilizando algoritmo albatros, la mayoría de las conexiones a lo largo de la diagonal de la matriz.

En la siguiente gráfica se muestra la evolución de la enfermedad a lo largo de los 100 días de simulación. El efecto que se puede observar del uso de este algoritmo para el muestreo del grafo es que, tras el primer pico de infección, en el grafo generado con albatros la enfermedad tiene un segundo pico mucho más alto que en el grafo generado por el algoritmo de escalado.

Esto indica que las estructuras que se forman en la conexiones favorecen el contagio de la infección al existir 'spreaders', o usuarios con gran cantidad de relaciones entre individuos y la conectividad existente, principalmente entre los diferentes grupos.

Características del grafo

En esta sección se describe el análisis realizado sobre la características que muestran los grafos resultantes de los algoritmos implementados (Escalado y Albatross) y la complejidad tanto temporal como espacial de ambos algoritmos.

En primer lugar se muestra el análisis del número de conexiones generadas y la dispersión de las matrices de adyacencia de los grafos generados por cada algoritmo.

En la siguiente gráfica se puede observar el número de conexiones con respecto al tamaño de la población.



Figura 31: Crecimiento de conexiones.

En la figura 31 se muestra el número de conexiones generadas por ambos algoritmos con respecto al tamaño de la población. En esta gráfica se puede observar como el número de

conexiones generadas por el algoritmo albatros tiene un comportamiento cuadrático mientras que en las conexiones generadas por el algoritmo de escalado su comportamiento es lineal.

Esto explica el comportamiento de la transmisión de la enfermedad entre los individuos siendo una mayor transmisibilidad cuanto mayor es el número de conexiones. De esta manera se puede explicar porque en grafos pequeños, en los que el algoritmo de escalado tiene más conexiones, se extiende a un menor número de individuos en el grafo generado por albatros.

Este efecto del crecimiento cuadrático indica que este algoritmo ajusta más al comportamiento real de las conexiones dentro de las redes sociales.

En la siguiente gráfica se puede observar la dispersión de la matriz de adyacencia de los grafos generados.

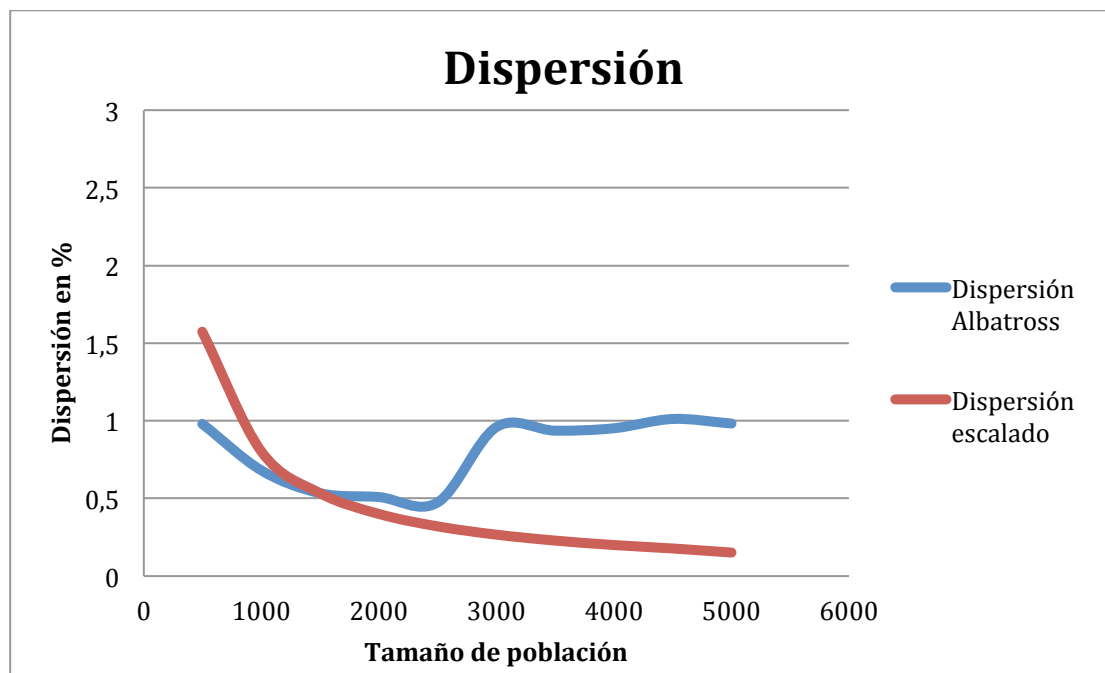


Figura 32: Dispersión de matrices.

Como se muestra en la gráfica de la figura 32, la dispersión de la matriz de adyacencia en el grafo generado por albatros tiende a mantenerse estable en torno al 1% de dispersión, es decir que de todas las relaciones posibles sólo se dan en un 1% de los casos. Sin embargo, la dispersión de los grafos generados por el algoritmo de escalado tiende a un comportamiento asintótico en 0 que conforme aumenta el número de individuos su dispersión se acerca más a 0.

Esto da lugar a que, en la propagación de la enfermedad, se transmita más rápido y a un mayor número de individuos en los grafos generados por el algoritmo Albatross conforme aumenta el tamaño de la población. Esto también da lugar a las diferentes olas de la enfermedad y la capacidad que tiene para mantenerse la enfermedad en el tiempo.

Complejidad del algoritmo

En este apartado se describe el análisis realizado de forma experimental tanto de la complejidad espacial como temporal de los algoritmo de muestreo.

Tras realizar este análisis se ha comprobado que tanto el coste tanto de memoria como de tiempo del algoritmo Albatross es muy superior al coste de ejecutar el algoritmo de Escalado.

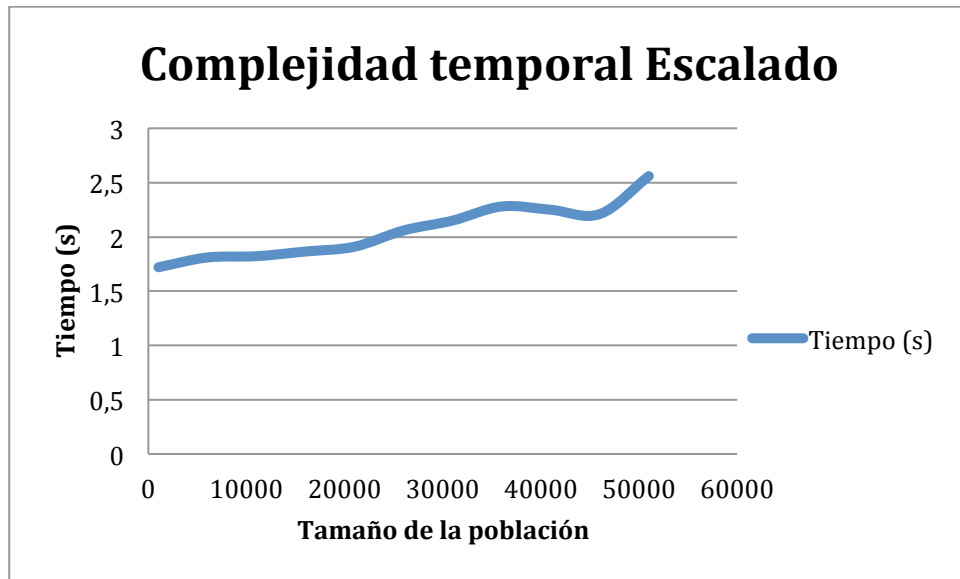


Figura 33: Complejidad temporal Escalado.



Figura 34: Complejidad temporal Albratross.

Tal como se observa en las gráficas se puede observar que el orden de complejidad temporal del algoritmo de escalado es de crecimiento lineal con relación al tamaño de entrada, mientras que, el algoritmo albatros, tiene un crecimiento exponencial de grado dos.

Esto implica que el tiempo de ejecución para poblaciones muy grandes crezca en gran manera en el algoritmo Albatross mientras que el tiempo necesario para ejecutar el algoritmo de escalado no requiera un tiempo significativo.

En cuanto al uso de memoria el comportamiento es muy similar al del tiempo. El algoritmo albatros presenta una complejidad espacial exponencial de segundo grado mientras

que la complejidad espacial del algoritmo de escalado es lineal con respecto al tamaño de entrada, en este caso la población del grafo.

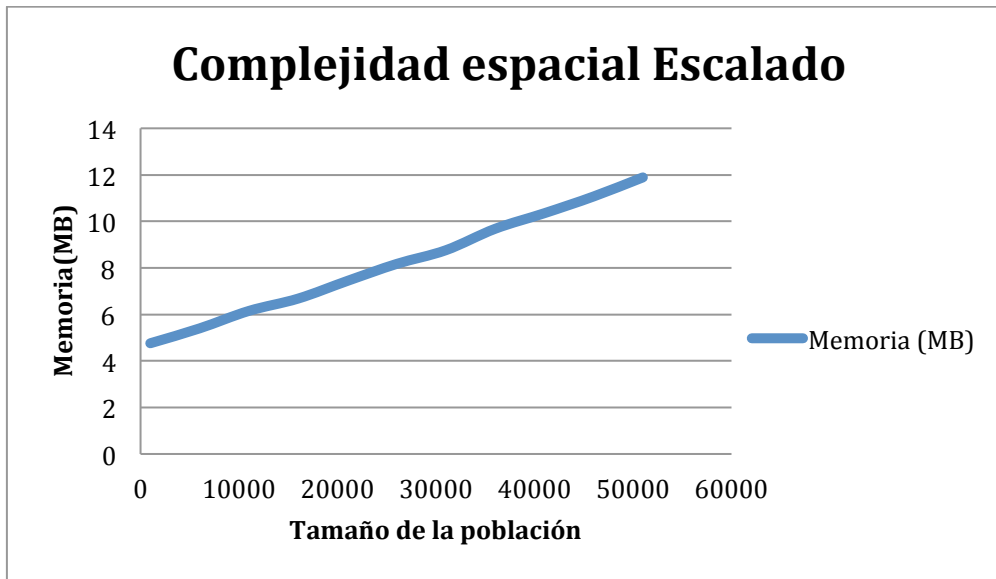


Figura 35: Complejidad espacial Escalado.



Figura 36: Complejidad espacial Albatross.

En definitiva, el algoritmo albatros, es mucho más complejo que el algoritmo de escalado y requiere de una mayor número de recursos. Sin embargo, los grafos que genera este algoritmo poseen características más similares a la realidad que los que genera el algoritmo de escalado.

6.2. Focos de infección en distintas ciudades.

En esta sección se describe el análisis realizado para los resultados de la simulación realizada partiendo de 3 infectados iniciales desde distintos puntos como foco inicial. En este

análisis se tendrán en cuenta los resultados numéricos obtenidos de la simulación de EpiGraph y la propagación analizada mediante la visualización gráfica de los mismos.

Para cada foco se realizarán cuatro simulaciones distintas con el fin de observar cómo varían las simulaciones dentro de un mismo foco y comparándolas con las de los distintos focos. Los focos que se tendrán en cuenta en esta evaluación serán Madrid, Barcelona, A Coruña y Leganés.

6.2.1. Madrid.

El primer conjunto de evaluaciones realizadas tienen como foco inicial la ciudad de Madrid con 3 infectados.

En primer lugar se ha analizado el número de infectados en cada instante de tiempo con el fin de observar el comportamiento de la enfermedad en cada una de las simulaciones. En la figura 37 se muestra la gráfica en la que se representa el número de infectados con respecto al tiempo de cada una de las ejecuciones realizadas.

Como se puede observar en la gráfica, el tiempo durante el que se produce la epidemia es muy similar y los picos de las diferentes olas de la infección se da entorno a puntos de tiempo muy similares.

Por otro lado, la primera oleada de menor tamaño se corresponde con la expansión de la enfermedad dentro de Madrid y una vez llegado al punto más alto de Madrid conforme la enfermedad disminuye en Madrid se expande al resto del país con gran velocidad. Esto es debido a la gran conectividad que presenta Madrid hacia el resto de ciudades de la simulación.

También se puede observar que en las situaciones en las que se dan los mayores picos de enfermedad se disipa a gran velocidad, como se observa en la serie temporal de la ejecución 1 de la figura 37, ya que, se desarrolla un mayor número de inmunes tras sufrir la enfermedad.

Debido a la inmunidad que se desarrolla en los individuos tras sufrir la enfermedad, estos individuos no hacen de transporte de la infección limitando la capacidad de propagación de la enfermedad. Esto implica que el número de infectados totales a lo largo de la simulación sea muy similar, como se puede observar comparando el área delimitada por línea de la función de cada ejecución.

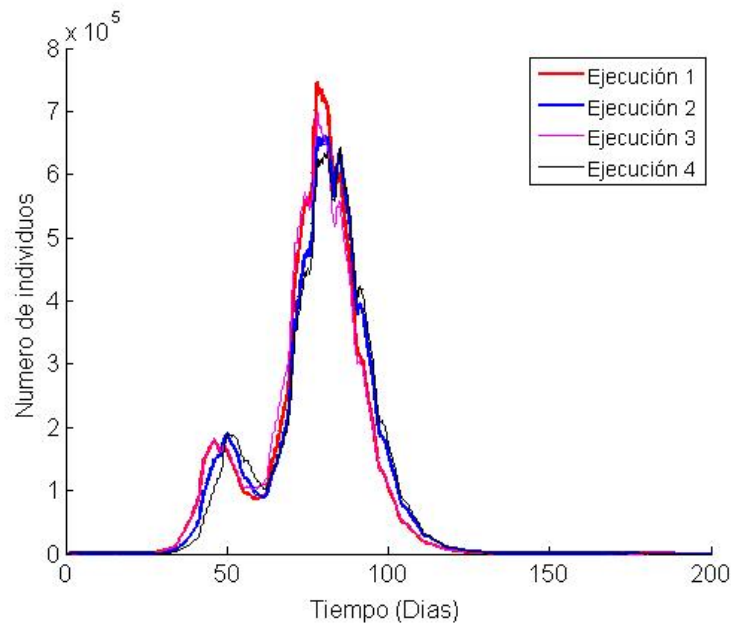


Figura 37: Gráfica de número de infectados Madrid.

Con el fin de poder observar más detalladamente el comportamiento de la función se muestra la tabla 49 con los valores numéricos de las características a evaluar de cada ejecución.

Ejecución	Máximo infectados	Duración	Total Infectados	Fallecidos
Primera ejecución	747244	129,65 días	8711830	13073
Segunda ejecución	661923	129,21 días	8721526	13160
Tercera ejecución	696590	128,31 días	8737378	13189
Cuarta ejecución	641955	127,22 días	8733970	13178

Tabla 49: Resumen ejecuciones de Madrid.

Como se puede observar el número de infectados totales es muy cercano lo que indica que la enfermedad se expande a un número aproximadamente idéntico de individuos, aunque de distinta manera. Puesto que la propagación de la enfermedad se basa en un proceso estadístico y debido al gran tamaño de las poblaciones, los valores de número total de infectados es muy similar.

En cuanto al número de fallecidos se puede observar que se trata de un valor proporcional al número de individuos que se infectan en la simulación, siendo mayor en las simulaciones en que se han infectado un mayor número de individuos. Esto es debido a que los fallecimientos vienen asociados a una probabilidad lo que implica que se mantenga una proporcionalidad de fallecidos con el número total de individuos infectados en todas las simulaciones.

Haciendo uso del visualizador se puede observar cómo se propaga la enfermedad y comprender en mayor medida el comportamiento de la curva de la enfermedad. Debido a la gran similitud entre las simulaciones con esta configuración se muestra únicamente la visualización de los resultados de una única simulación.

En torno al día 50 haciendo uso de la herramienta desarrollada, se puede observar que se da el punto más alto de infección de Madrid que se corresponde con el primer pico de la gráfica mostrada anteriormente. De esta manera se puede confirmar que esta primera oleada se corresponde con el momento en que Madrid se encuentra en el nivel más alto de infección.

En este mismo momento comienza a expandirse a otras ciudades que tienen un nivel bastante bajo de infección.



Figura 38: Día 45 simulación de Madrid.

Pasado este punto la infección en Madrid comienza a remitir conforme aumenta la infección en el resto de ciudades de España. En un primer lugar hay menos casos de infectados durante el tiempo que empieza a darse los primeros casos en las otras ciudades y posteriormente empiezan a expandirse dentro de éstas.

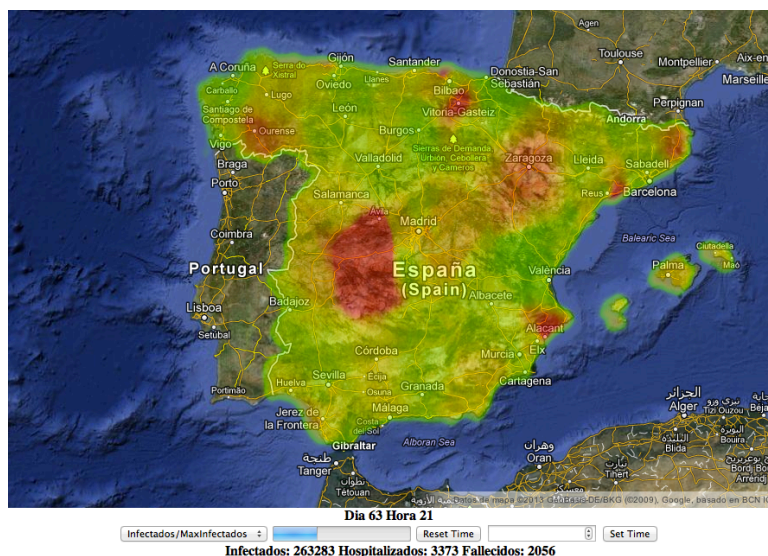


Figura 39: Día 63 simulación de Madrid.

El momento en que comienza a expandirse la enfermedad dentro de las ciudades contagiadas por Madrid es cuando la gráfica mostrada anteriormente tiene una tendencia creciente hasta llegar al punto máximo de la enfermedad.

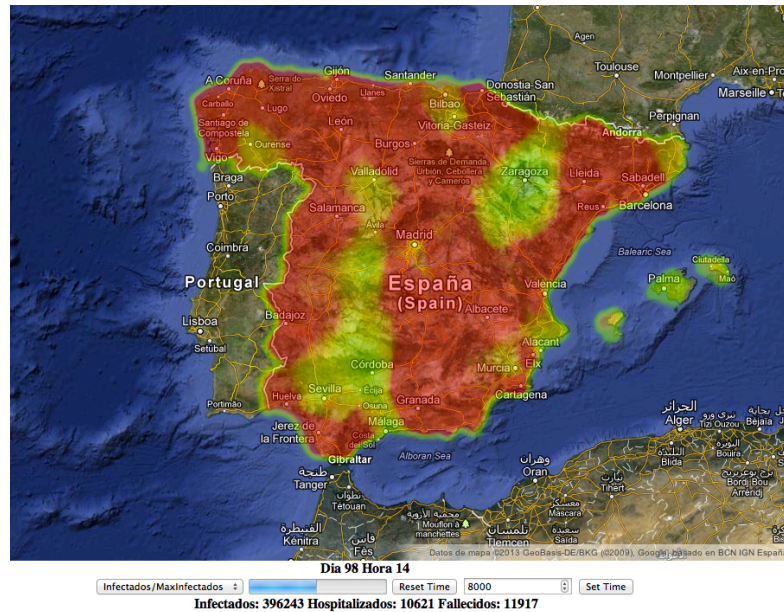


Figura 40: Día 98 simulación de Madrid.

Una vez superado este nuevo pico, se puede observar en la visualización como empieza a remitir la enfermedad empezando por las primera ciudades infectadas y aquellas cuya expansión de la enfermedad se ha dado a un mayor ritmo. Esto es debido a que aquellos individuos que superan la enfermedad desarrollan anticuerpos y ya no actúan como medio de transporte para la enfermedad.

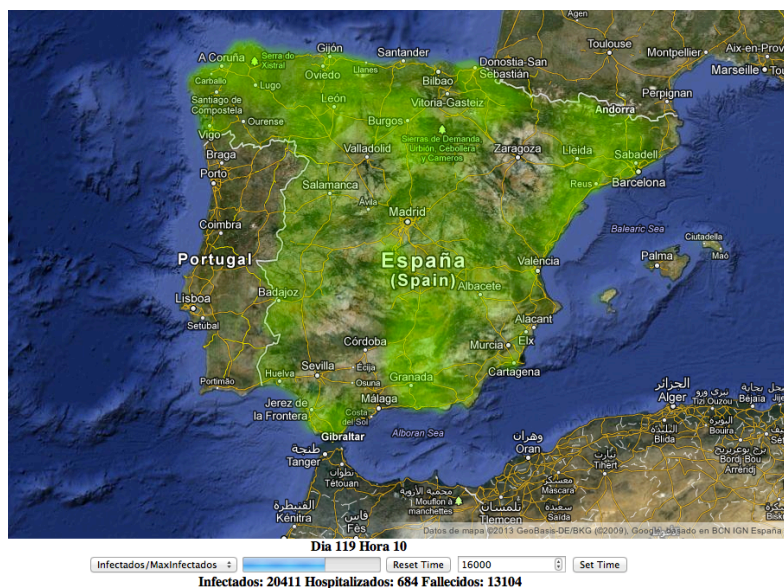


Figura 41: Día 119 simulación de Madrid.

Finalmente, se observa como la epidemia finaliza disminuyendo progresivamente el número de infectados en todas las ciudades del país hasta desaparecer por completo la enfermedad. Las pequeñas diferencias, existentes en las curvas de las diferentes simulaciones, se deben a la aleatoriedad de la selección de individuos del simulador.

6.2.2. A Coruña.

En esta segunda evaluación se ha tomado como foco inicial de la epidemia la ciudad de A Coruña.

En la figura 42 se muestra una gráfica donde se puede comparar el comportamiento de la enfermedad con respecto al tiempo y ver el número de infectados totales dentro del país.

En primer lugar se puede observar en la gráfica que el tiempo que tarda en comenzar a expandirse es distinto en las simulaciones. Esto es debido a las diferentes ciudades a las que se transmite la enfermedad. En las situaciones en las que la enfermedad se expande más rápido son simulaciones en las que las ciudades más conectadas se infectan antes, lo que implica un crecimiento mayor.

Al contrario que pasa en Madrid, que es una ciudad de mayor tamaño y por tanto transmite la enfermedad a las ciudades vecinas por igual, la población de A Coruña es mucho menor y esta menos conectada da lugar a que la enfermedad se transmita en distinta medida al resto de ciudades.

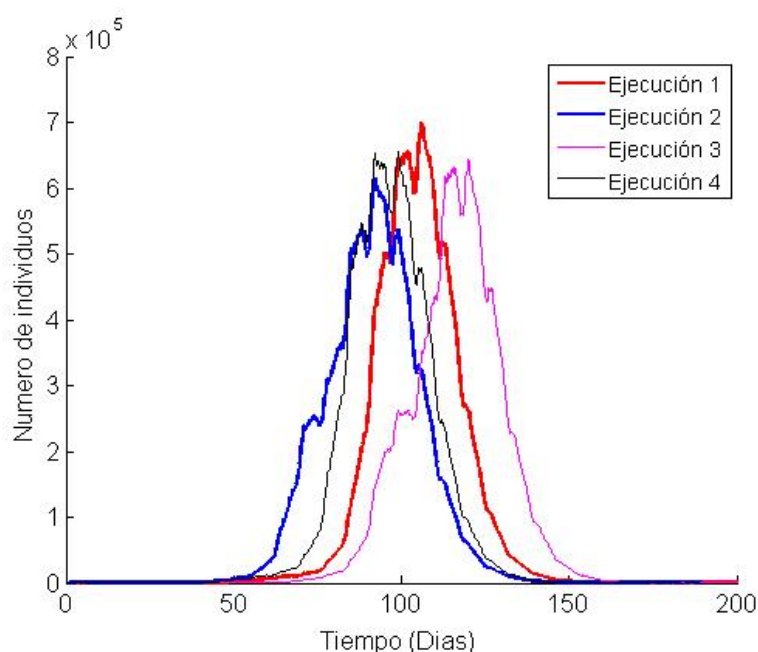


Figura 42: Gráfica de número de infectados A Coruña.

Con el fin de facilitar la observación de las características que definen el comportamiento de la enfermedad se han analizado los datos que se muestran a continuación en la siguiente tabla.

Ejecución	Máximo infectados	Duración	Total Infectados	Fallecidos
Primera ejecución	700399	136,77 días	8722716	13417
Segunda ejecución	615338	134,77 días	8730685	13216
Tercera ejecución	644042	136,01 días	8701591	13214
Cuarta ejecución	654236	136,61 días	8728374	13817

Tabla 50: Resumen ejecuciones de A Coruña.

Como se puede observar en la tabla 50 hay una variación significativa en el pico más alto, aunque, al igual que ocurría en la simulación sobre Madrid, el número de infectados totales en la epidemia es muy similar. Esto nos indica que el número de infectados totales prácticamente no depende del foco donde se inicia la enfermedad, aunque, si implica variaciones en la duración y el número de infectados en el pico máximo de la enfermedad.

Al igual que se comprobó en las simulaciones realizadas iniciando la enfermedad en Madrid, el número de fallecidos totales mantiene la relación con el número de individuos infectados a lo largo de la simulación.

A continuación, haciendo uso del visualizador, se busca analizar como se expande la enfermedad por las diferentes ciudades con el fin de comprender con mayor claridad las diferencias existentes en las diferentes gráficas.

En los primeros instantes de tiempos se puede observar como se da un crecimiento relativamente lento dentro de la ciudad de A Coruña. Una vez se alcanza niveles altos en el porcentaje de infectados se puede observar como se transmite al resto de ciudades.

Debido a que no es ciudad tan interconectada como Madrid se puede observar como la expansión es muy gradual y va avanzando a las distintas ciudades lentamente.

Aunque se hace difícil de observar en la gráfica, debido a que la población de A Coruña no es muy elevada, entorno al día 50 se da el pico de máxima infección en todas las simulaciones en la ciudad de A Coruña.

A partir de este punto, dependiendo de un factor de aleatoriedad en las ciudades destino de los transportes entre ciudades, la enfermedad va expandiéndose a diferentes ciudades dependiendo de la simulación.

Dependiendo de las ciudades que se van infectando, la velocidad a la que se expande la enfermedad por el país varía significativamente. Esto es debido a que las infecciones en ciudades pequeñas y conectadas en menor medida tienen más dificultad para expandir la enfermedad a otras ciudades.

En la primera de las ejecuciones se puede observar como las primeras ciudades contagiadas no son demasiado grandes por lo que se da un crecimiento lento en la gráfica en los primeros días de la infección.



Figura 43: Día 66 primera simulación A Coruña.

Sin embargo, a partir del día 75 aproximadamente la propagación de la enfermedad experimenta un crecimiento acelerado debido al contagio de Madrid. Al ser la capital y ser una ciudad más interconectada con el resto favorece en gran medida la expansión a las ciudades no infectadas. Debido a que en esta simulación las ciudades infectadas son pequeñas y no son demasiadas se ve un crecimiento aún mayor que en el resto de simulaciones.

En las dos siguientes figuras se muestra el estado de la enfermedad en el momento de contagio de Madrid y tras su posterior expansión al resto de ciudades. Estas imágenes coinciden con el punto de mayor crecimiento y el pico más alto de la enfermedad.

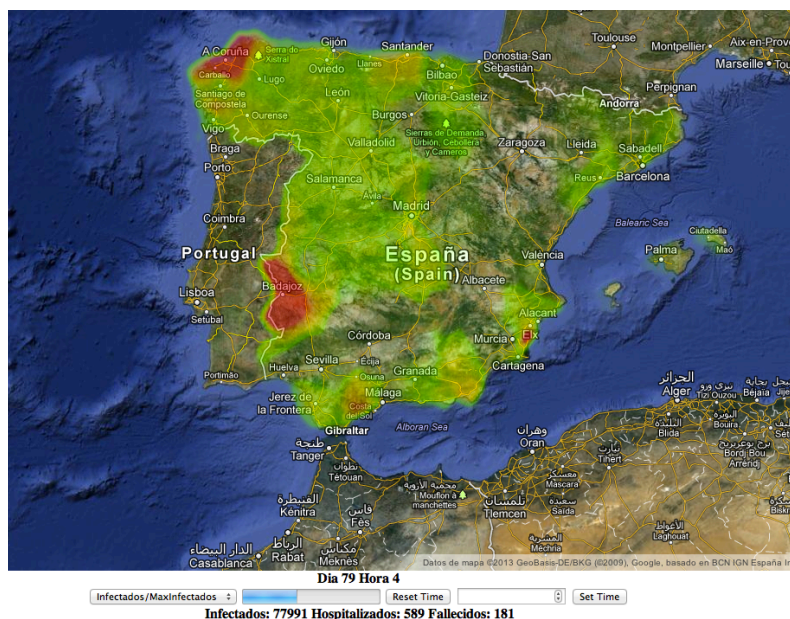


Figura 44: Día 79 primera simulación A Coruña.

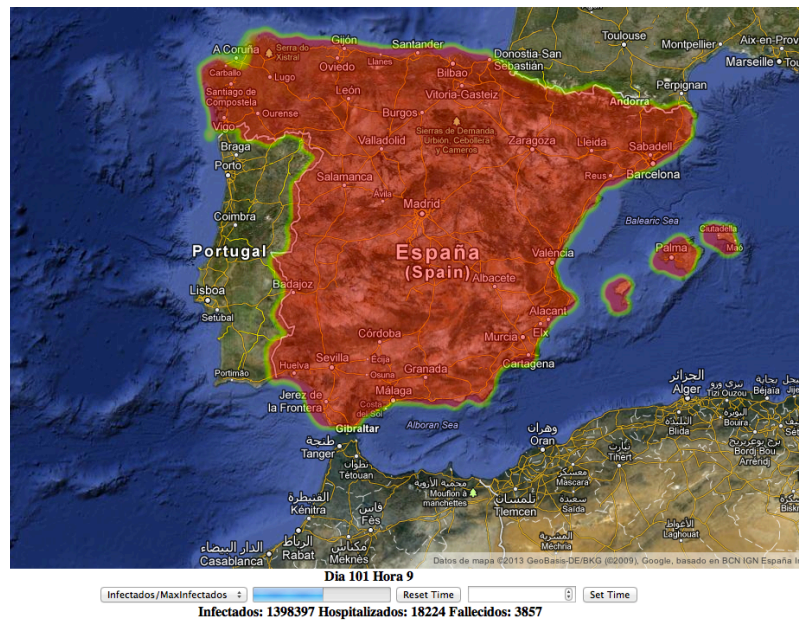


Figura 45: Día 101 primera simulación A Coruña.

Este hecho implica a su vez que, posteriormente al pico de máxima infección, la enfermedad decaiga a una mayor velocidad, ya que, la gran mayoría de las ciudades se han infectado prácticamente en el mismo momento y por tanto desaparece casi simultáneamente de todas ellas.

En la figura 46 se muestra como desaparece la enfermedad casi uniformemente en todas las ciudades de la simulación.



Figura 46: Día 141 primera simulación A Coruña.

Por otro lado en la segunda ejecución se puede observar como el crecimiento al inicio de la expansión es muy superior al crecimiento experimentado en el resto de simulaciones. En la

siguiente figura se muestra el estado de la enfermedad en ese punto y puede verse como las ciudades a las que se expande, al contrario que en la primera simulación, son ciudades más importantes y con una mayor población.

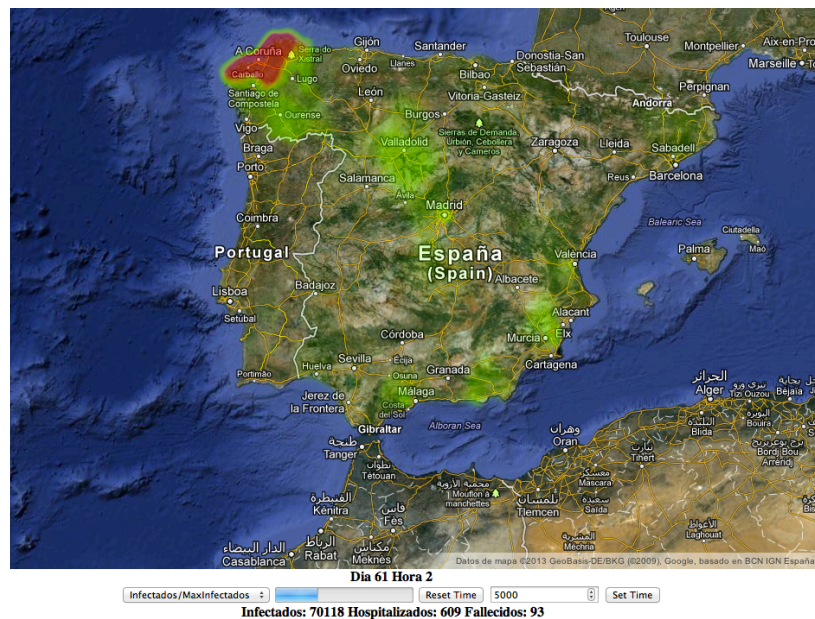


Figura 47: Día 61 segunda simulación A Coruña.

En esta simulación se infectan al comienzo las ciudades de Madrid y Barcelona lo que implica que, a partir del punto en que la enfermedad se encuentra desarrollada en estas ciudades, se transmita con mayor facilidad hacia otras ciudades que no ha sido infectadas. Debido a esto el punto de máxima infección se da en un tiempo muy inferior al resto de simulaciones que tardan más en llegar a estas ciudades.

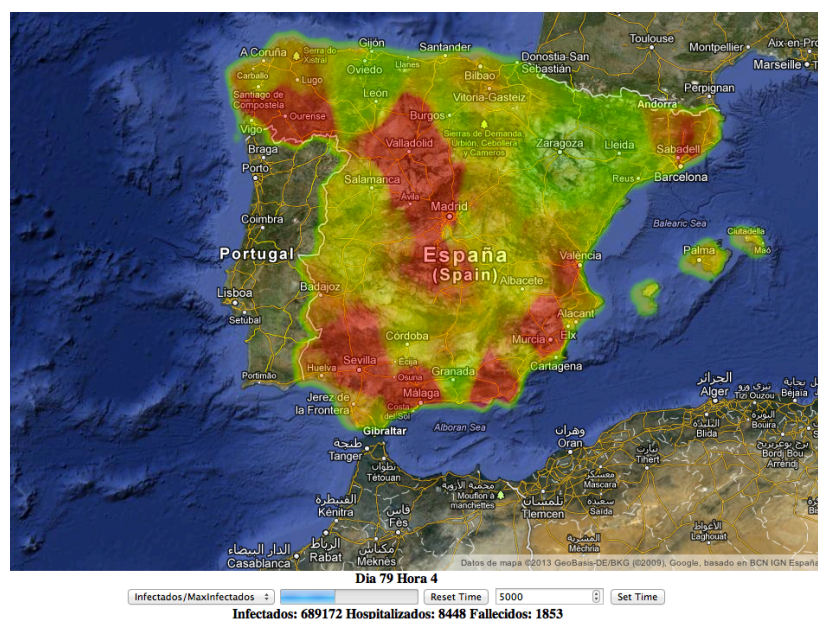


Figura 48: Día 79 segunda simulación A Coruña.

Por otro lado debido a la aleatoriedad en la selección de los individuos que viajan de una ciudad a otra también se observa un crecimiento más progresivo ya que no se transmite al mismo tiempo a todas las ciudades. Esto es debido a que al mismo tiempo que se infectan Barcelona y Madrid se infectan un número superior de ciudades.

De igual manera la desaparición de la enfermedad se da menos bruscamente conforme se va superando la enfermedad en las distintas ciudades.

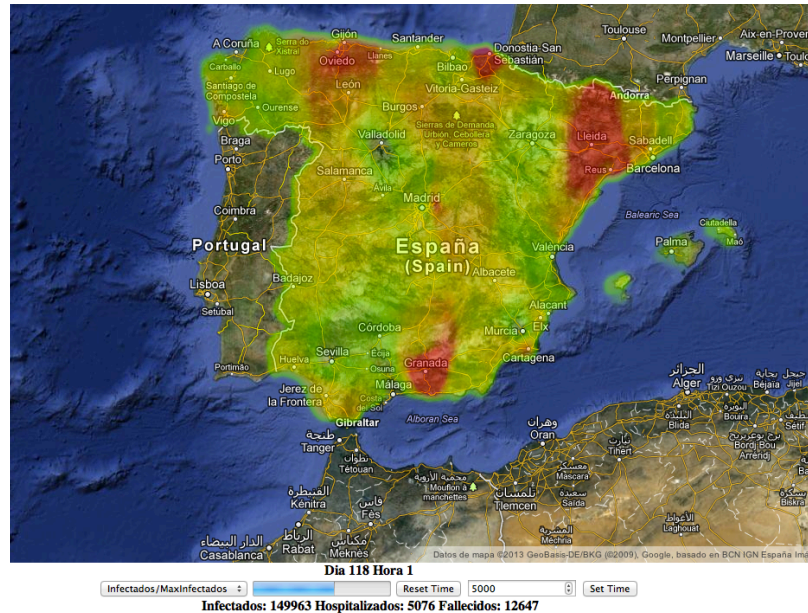


Figura 49: Día 118 segunda simulación A Coruña.

En cuanto a las otras dos simulaciones, las diferencias dependen únicamente de las ciudades que se infectan antes o después.

En el caso de la cuarta simulación, el crecimiento rápido que experimenta es por el mismo motivo que en la primera simulación. Sin embargo, en este caso la enfermedad llega antes a Madrid lo que implica que el contagio masivo se dé en un instante de tiempo anterior.

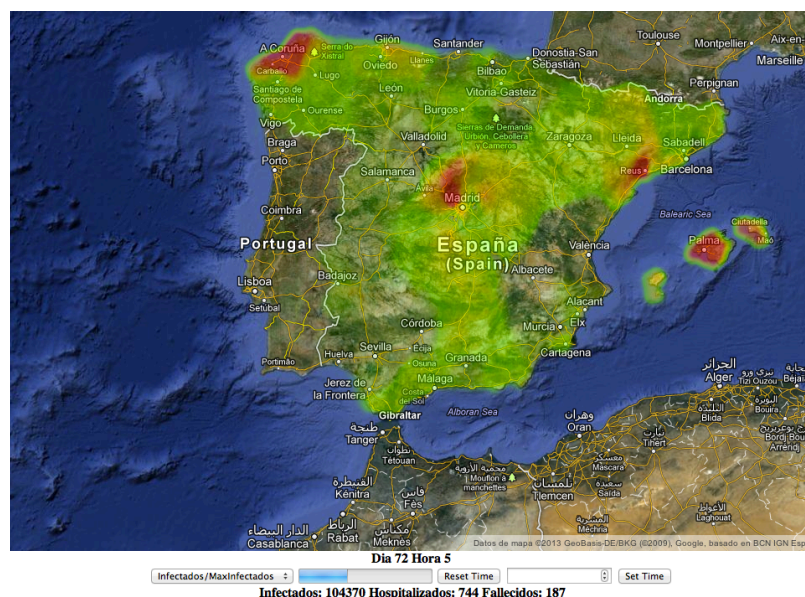


Figura 50: Día 72 cuarta simulación A Coruña.

Igual que ocurría con la primera ejecución del simulador, en esta simulación la infección se expande desde Madrid casi simultáneamente a todas las ciudades del país, por ello se da un crecimiento muy pronunciado en el número de infectados.

Por otro lado, en la tercera simulación ocurre el caso contrario, las ultimas ciudades en ser infectadas son las más importantes (Madrid, Barcelona, ...) por este motivo el momento en que se da la expansión más rápida de la enfermedad no llega hasta bastante tarde con relación al resto de simulaciones.

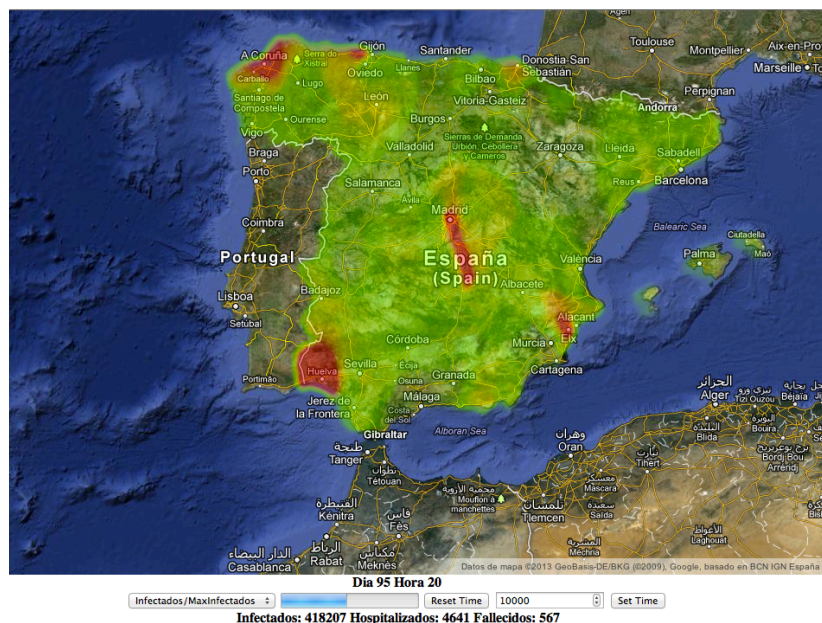


Figura 51: Día 95 tercera simulación A Coruña.

A partir de este punto la simulación tiene un comportamiento similar al de la segunda ejecución.

6.2.3. Barcelona.

En este bloque de evaluaciones se ha simulado la enfermedad partiendo de la ciudad de Barcelona.

Los resultados de las 4 ejecuciones son muy similares a los obtenidos para la ciudad de Madrid. Esto es debido a que ambas ciudades tienen una relación muy similar con el resto de ciudades y una gran población.

Por este motivo, al igual que en Madrid, se produce una primera oleada de la enfermedad en torno al día 40 cuando se alcanza el máximo nivel de contagio dentro de la ciudad de Barcelona. En esta situación este pico es de un tamaño inferior al resultante en Madrid debido a la diferencia de población existente entre ambas ciudades.

Una vez se ha infectado Barcelona, la enfermedad se propaga con bastante facilidad al resto de ciudades y estas se infectan en un pequeño periodo de tiempo lo que da el segundo pico de la enfermedad. Puesto que la gran mayoría de ciudades se infectan prácticamente al

mismo tiempo, se da un gran pico cuando todas ellas alcanzan sus máximos locales de la infección.

Por otro lado, aunque la conectividad existente entre las ciudades de Madrid y Barcelona son muy similares, la ciudad de Madrid tiene una mayor conectividad al ser la capital del país y situarse en el centro de la península. Esto hace que en Madrid las simulaciones sean más similares, ya que, al tener una conectividad tan alta la propagación al resto de ciudades sea en un periodo de tiempo menor que en Barcelona.

En Barcelona, al ser este periodo de propagación al resto de ciudades algo mayor que en Madrid, la enfermedad puede propagarse a tiempos ligeramente distintos a las distintas ciudades.

En la figura 52 se muestra la gráfica del número de infectados con respecto al tiempo de las simulaciones realizadas partiendo de Barcelona.

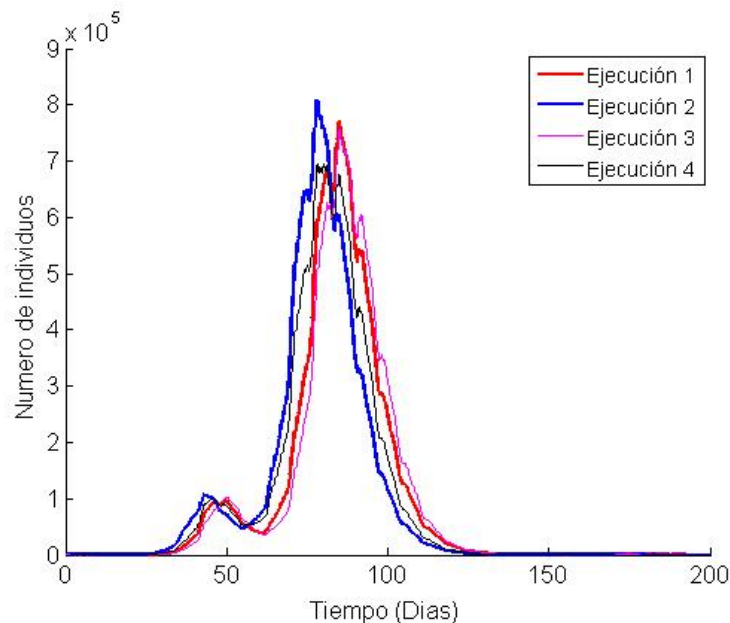


Figura 52: Gráfica de número de infectados Barcelona.

En la siguiente tabla se muestra el resumen de resultados de la ejecución con el fin de analizar con mayor precisión la propagación de la enfermedad.

Ejecución	Máximo infectados	Duración	Total Infectados	Fallecidos
Primera ejecución	772142	132.28	8713183	13335
Segunda ejecución	808093	128.86	8721354	13376
Tercera ejecución	756465	132.35	8704130	13223
Cuarta ejecución	694045	130.61	8705685	13199

Tabla 51: Resumen ejecuciones de Barcelona.

Como se puede observar en la tabla 51 el número total de infectados a lo largo de la simulación se mantiene sin demasiadas variaciones con respecto al resto de simulaciones. Esto es debido a que la enfermedad se propaga siguiendo las probabilidades de contagio del simulador. Por este motivo al calcular el porcentaje del total de infectados con respecto a la población total de la simulación se mantiene en torno al 40% con pequeñas variaciones. Estas

variaciones se deben a que se trata de un sistema estocástico y por tanto tiene un factor de aleatoriedad.

Por otro lado, el pico máximo de infectados difiere en gran medida dependiendo del foco inicial. En las ciudades más interconectadas, como Barcelona o Madrid, estos picos son muy superiores a las ciudades menos conectadas como es el caso de A Coruña.

Debido a la gran similitud entre las cuatro evaluaciones solo se realizara el análisis de una de las ejecuciones utilizando el visualizador con el fin de analizar con mayor exactitud la propagación de la enfermedad.



Figura 53: Día 47 simulación Barcelona.

En la figura 53 puede observarse el punto de la simulación en que Barcelona alcanza su máximo nivel de infección que se refleja con el color rojo sobre la capital de Cataluña. Como se puede observar en la gráfica, pasado este momento empieza a decaer la enfermedad en Barcelona y comienza a propagarse hacia el resto de ciudades.



Figura 54: Día 58 simulación Barcelona.

En la figura 54 puede observarse como debido al gran nivel de interconexión del resto de ciudades con Barcelona la enfermedad se propaga simultáneamente a una gran mayoría de las ciudades del país. Al igual que ocurría en la simulación cuyo foco inicial se situaba en Madrid, el hecho de que la enfermedad se propague rápidamente al resto de ciudades da lugar a un mayor crecimiento en el número de infectados y provoca un máximo superior al de las simulaciones realizadas sobre ciudades menos interconectadas como A Coruña.

Este pico máximo del número de infectados queda reflejado en la figura 55 donde se ve que la mayor parte del país queda representado con el color rojo que representa el máximo de infección local.

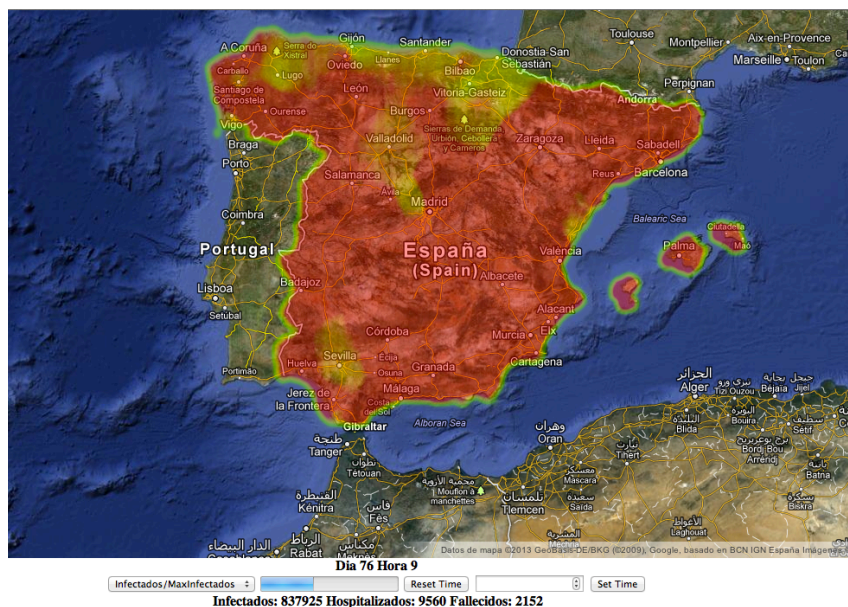


Figura 55: Día 76 simulación Barcelona.

Una vez superado este punto de máxima infección la enfermedad pasa a un proceso de remisión hasta ser superada la enfermedad en todo el país.

6.2.4. Leganés.

Para la última evaluación con los diferentes focos de partida se ha tomado como ciudad de inicio de la epidemia Leganés.

Al ser Leganés una ciudad cercana a Madrid, está muy bien conectada con el resto de ciudades de la simulación. Esto produce que el contagio se expanda rápidamente a otras ciudades y, por tanto, la epidemia se expanda a gran velocidad dando picos muy similares a los que se dan en la segunda oleada en Madrid.

La diferencia existente entre la forma de la función de esta simulación con respecto a las simulaciones de Madrid radica en que la primera oleada no es tan significativa ya que la población de Leganés es muy inferior a la de la capital.

Aunque no es apreciable en la siguiente función en torno al día 25 de la simulación se da el pico en Leganés. Como ya se ha comprobado en el resto de simulaciones, el pico de máxima

infección sobre el foco inicial supone el comienzo de la escala masiva al resto de ciudades. Esto viene dado debido a que, al existir una mayor cantidad de infectados en la ciudad la probabilidad de que viaje un individuo infectado a otras ciudades aumenta en gran medida.

Por otro lado, con respecto a Madrid, la enfermedad llega a su máximo nivel de contagio en un tiempo inferior. Esto es debido a que el tamaño de la población de Leganés es menor y la probabilidad de seleccionar un individuo infectado para realizar un viaje a otra ciudad es muy superior.

Una vez infectadas el resto de ciudades la expansión de la enfermedad se transmite a una mayor velocidad hasta llegar a su valor máximo.

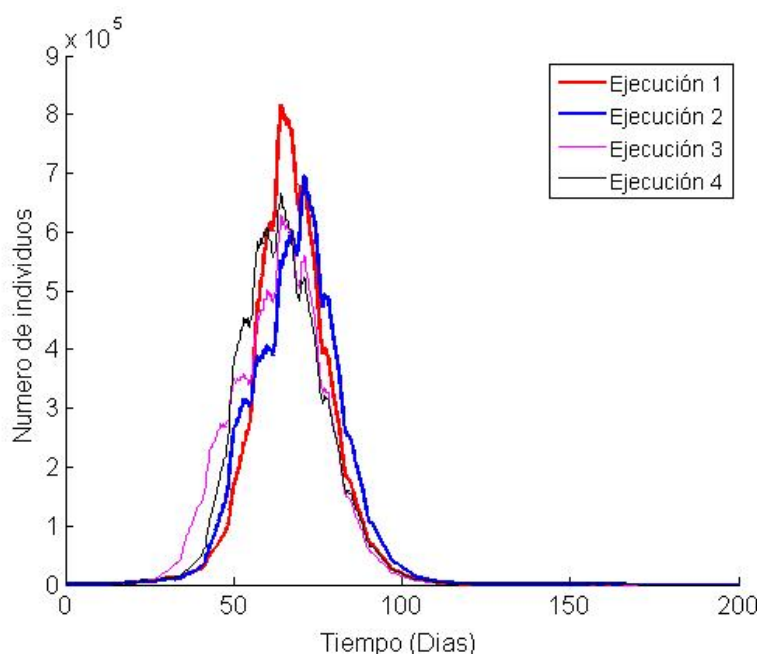


Figura 56: Gráfica de número de infectados Leganés.

En la tabla 52 se pueden observar el resumen de los datos de las simulaciones:

Ejecución	Máximo infectados	Duración	Total Infectados	Fallecidos
Primera ejecución	815410	132,26 días	8730925	13029
Segunda ejecución	694659	135,38 días	8713021	13283
Tercera ejecución	628975	130,22 días	8729571	13176
Cuarta ejecución	665172	132 días	8729340	13235

Tabla 52: Resumen ejecuciones de Leganés.

Observando los datos puede verse como el número de infectados totales es muy similar en todas las ejecuciones. Sin embargo, en la primera de ellas el número de infectados es ligeramente superior. Esta diferencia viene debida a la probabilidad aleatoria de contagio.

En cuanto al número máximo de infectados en un instante de tiempo los valores varían en algunos casos en gran medida. Esto es debido a cuales son las primeras ciudades hacia las que se propaga la enfermedad. En la primera simulación las primeras ciudades infectadas son ciudades muy interconectadas con el resto lo que supone un pico más alto como ocurre en las simulaciones de Madrid o Barcelona.

Sin embargo en el resto de simulaciones las ciudades infectadas no son tan importantes, esto da lugar a una propagación más similar a la propagación de la enfermedad en A Coruña. Por otro lado, al igual que en el resto de simulaciones, tanto la duración como el número de fallecidos tienen valores muy similares.

Con el fin de comprender mejor los diferentes resultados de las simulaciones se ha comprobado el comportamiento de la enfermedad haciendo uso de la herramienta de visualización desarrollada.

En los primeros días de la infección la propagación de la enfermedad en todas las simulaciones es similar. La enfermedad se reproduce en Leganés hasta alcanzar su pico más alto y a partir de ahí comienza a expandirse al resto de ciudades.



Figura 57: Día 27 simulación Leganés.

En este momento es donde se diferencian las simulaciones realizadas dependiendo de las ciudades que han sido las primeras en contagiarse. Si las primeras ciudades contagiadas tienen un alto grado de conectividad con otras ciudades y el tamaño de estas poblaciones es suficientemente grande, los picos resultantes son más pronunciados y alcanzan valores mayores.

Si por otro lado las ciudades son más pequeñas y/o menos conectadas el crecimiento que experimenta la enfermedad es menos pronunciado y se dan pequeños picos. Estos pequeños picos del número de infectados se corresponden con las bajadas en el porcentaje de infectados en las ciudades infectadas.

Al darse una expansión más lenta las diferentes ciudades se infectan en tiempos más espaciados y no tan simultáneamente. Esto implica que las primeras ciudades infectadas comiencen a superar la enfermedad antes y, por este motivo, el pico de máximo número de infectados sea menor y se den pequeños picos a lo largo de la expansión de la enfermedad por el territorio.

Esto puede observarse claramente comparando la primera y la tercera ejecución de esta simulación. En la primera de ellas, como se muestra en la figura 58, las ciudades que se infectan primero son grandes ciudades muy interconectadas, como son Madrid y Barcelona. A partir de ahí comienza el contagio acelerado que se observa en la gráfica.



Figura 58: Día 40 primera simulación Leganés.

En el caso de la tercera ejecución, aunque se contagia a en un menor tiempo, las ciudades que se infectan no tienen la misma importancia que las infectadas por la primera ejecución. En la siguiente imagen se puede observar que en este caso Madrid y Barcelona se infectan más tarde aunque si se infectan las ciudades de su periferia.

Aunque abarca más territorio la enfermedad en el mismo instante de tiempo, las zonas infectadas están más despobladas y por tanto no implican un pico tan elevado como en la primera ejecución.

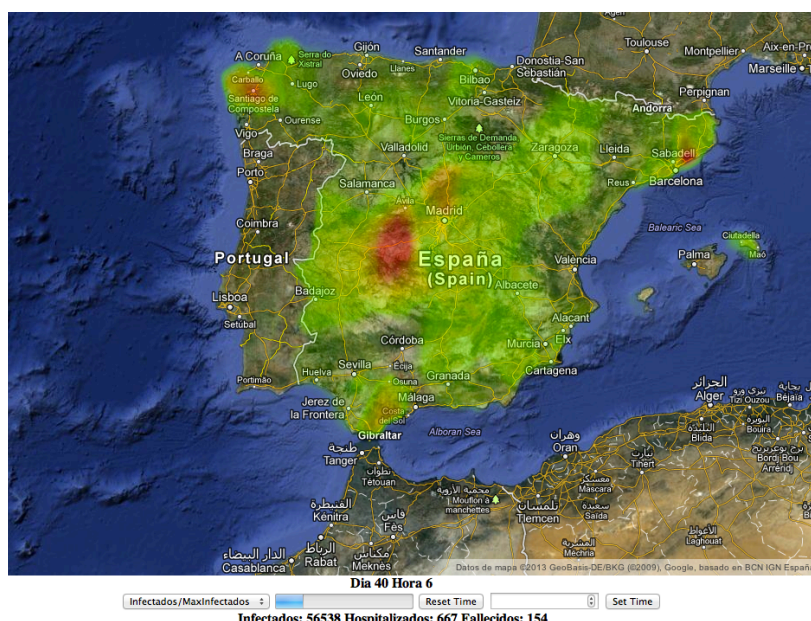


Figura 59: Día 40 tercera simulación Leganés.

Por otro lado el hecho de que las primeras ciudades contagiadas no estén tan infectadas hace que la expansión se de en tiempos distintos, al contrario de lo que pasa en la primera ejecución.

Debido a esto, a la vez que se van infectando nuevas ciudades, las infectadas anteriormente comienzan a superar la enfermedad. Esto da lugar a los diversos picos que se dan a lo largo del crecimiento de la epidemia.

En la figura 60 se puede observar la diversidad de situaciones en el estado de la enfermedad en las distintas ciudades. Los diversos colores indican que las ciudades se encuentran en distintos puntos de contagio interno de la ciudad. Algunas ciudades como Santiago ya han alcanzado el punto máximo de la enfermedad mientras que otras, como Córdoba o Bilbao, no se han contagiado todavía o tienen un nivel bajo de porcentaje de infectados.

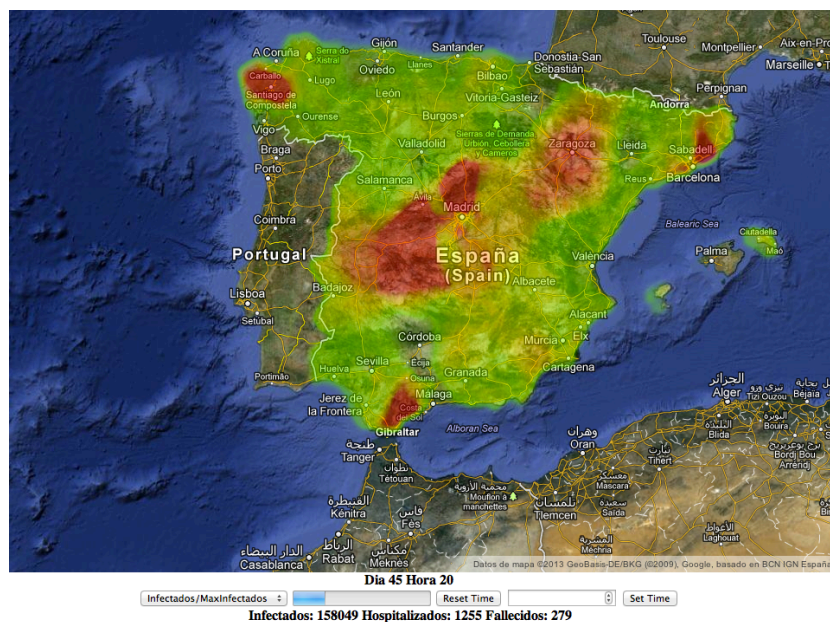


Figura 60: Día 45 tercera simulación Leganés.

Una vez las ciudades que han alcanzado el pico máximo comienzan a superar la enfermedad, mientras que en las ciudades que comenzaba la infección alcanza su pico máximo. Por ello se dan puntos en los que el descenso de infectados, en las ciudades que comienzan a superar la enfermedad, supera al aumento en las ciudades infectadas posteriormente. Esto se refleja en las gráficas como pequeños picos de descenso durante el contagio de la enfermedad.

6.3. Evaluación de distintos valores de R_0 .

En esta sección se ha evaluado el efecto que tiene sobre la epidemia la variación del valor de la tasa de reproducción básica o neta (R_0). Este valor representa el número de descendientes en promedio de cada individuo infeccioso a lo largo de su vida. De esta manera se puede comprobar el efecto de enfermedades con mayor o menor virulencia.

Para realizar esta evaluación se utilizan los valores de R_0 de 0.5, 1, 1.5 y 3.

R0 = 1.5

En la figura 61, se muestran los resultados obtenidos de las simulaciones realizadas aumentando el valor de R0 a 1.5. Como se puede observar el comportamiento de la enfermedad es similar con la diferencia de que el número de infectados es ligeramente superior y el tiempo en que se mantiene activa la enfermedad se reduce.

Esto es debido a que el valor de R0 define la velocidad de propagación de la enfermedad. Al aumentarse este valor la velocidad de propagación es más rápida lo que implica que la propagación, tanto dentro del foco inicial como al resto de ciudades, se da a una mayor velocidad.

Debido a este motivo la enfermedad se mantiene en un mayor número de individuos y se propaga en mayor medida. Debido a que el número de contagiados por infectado es mayor, antes de superar la enfermedad y que las conexiones que tiene el individuo dejen de hacer de transporte a la enfermedad, la enfermedad se propaga a un mayor número de colectivos y subconjuntos de individuos. Esto da lugar a que el número de infectados sea bastante mayor.

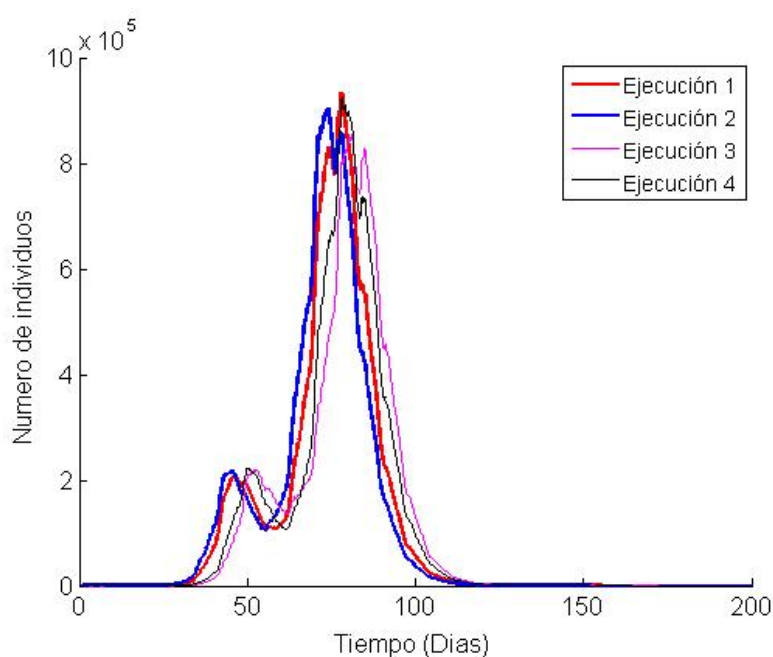


Figura 61: Epidemia con R0=1.5.

Como se puede observar en la tabla 53 el número de infectados totales al aumentar el valor de R0 disminuye el tiempo en que se encuentra activa la enfermedad en gran medida. Esto es debido a que dada la rápida propagación de la enfermedad se desarrolla rápidamente en todas las ciudades y, a su vez, se supera en todas las ciudades antes de producirse más contagios.

Una vez se ha llegado al pico de máxima infección y los individuos comienzan a superar la enfermedad y desarrollan inmunidad a la misma los enlaces que de estos individuos dejan de servir de transporte de la enfermedad al no poder ser individuos que permitan el contagio. Esto

implica que las posibilidades de transmitirse la enfermedad se reduzcan rápidamente al existir en este momento gran cantidad de individuos inmunes, lo que da lugar a una rápida desinfección.

Ejecución	Máximo infectados	Duración	Total Infectados	Fallecidos
Primera ejecución	933169	117,26 días	9796352	14894
Segunda ejecución	904723	114.40 días	9812209	14902
Tercera ejecución	855620	114.17 días	9806115	14976
Cuarta ejecución	924681	115.13 días	9807418	14979

Tabla 53: Resumen evaluación con $R_0=1.5$

Si observamos la simulación en el visualizador y la comparamos con las simulaciones realizadas anteriormente sobre Madrid con un valor de R_0 de 1.3 se puede observar que en el mismo instante de tiempo, el día 45, en la simulación inicial prácticamente se encuentra infectada Madrid únicamente. Sin embargo, en las ejecuciones con el valor de R_0 a 1.5 la enfermedad se ha propagado a una mayor cantidad de ciudades. Esto es debido a que se alcanza a una mayor velocidad el máximo local de la infección de Madrid y además con un mayor número de infectados lo que favorece que se propague al resto de ciudades.



Figura 62: Día 45 simulación $R_0=1.5$.

A partir de este momento las diferencias aumentan en mayor medida en cuanto al estado de la enfermedad en un mismo instante de tiempo. Si observamos el estado en la simulación con R_0 a 1.3, en el día 63, el estado de la enfermedad se esta iniciando en todas las ciudades mientras que en estas simulaciones con R_0 a 1.5 la enfermedad se encuentra bastante desarrollada en gran parte de las ciudades.

Aunque en el estado del día 45 la diferencia no eran tan significativas, la propagación dentro de las ciudades se da a una mayor velocidad con este R_0 lo que implica que estas ciudades experimente una propagación mucho mas alta. En al figura 63 se puede observar como se encuentran infectadas aproximadamente el 50% de las ciudades a un nivel alto de infección.

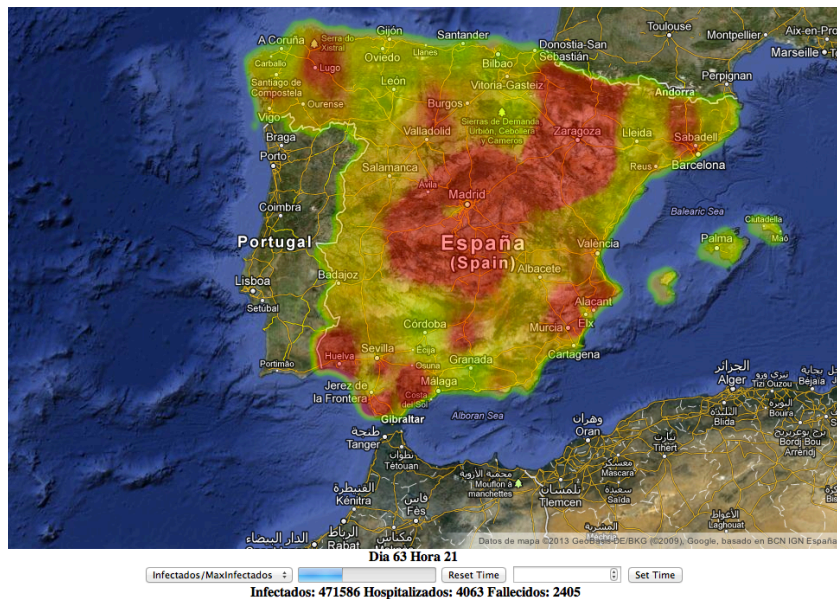


Figura 63: Día 63 simulación $R_0=1.5$.

Igualmente el hecho de que la enfermedad se haya propagado a este gran número de ciudades y que dentro de ellas la enfermedad se propague a una gran velocidad implica que el punto máximo de la infección, además de ser mas alto, se de en un instante de tiempo anterior. Si comparamos el día 98 de la simulación podemos observar como en esta ejecución la enfermedad esta siendo superada la infección, como se aprecia en la figura 64, mientras que en la ejecución con R_0 a 1.3 la enfermedad no ha terminado de propagarse al todo el país.

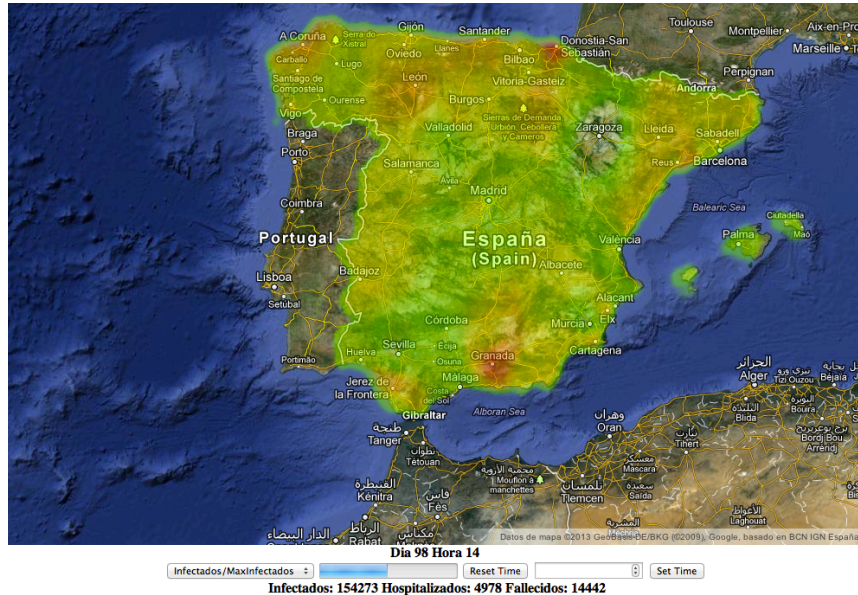


Figura 64: Día 98 simulación $R_0=1.5$.

R0 = 3

En las siguientes simulaciones se ha aumentado el valor de R0 a un valor de 3 con el objetivo de observar este aumento hasta que punto influye sobre la enfermedad. Como se ha observado en las simulaciones anteriores la velocidad de propagación que implica el valor de R0 se traduce en un mayor número de infectados y un tiempo de actividad de la enfermedad menor.

En esta simulaciones se ha comprobado que el efecto de aumentar el valor de R0 provoca las diferencias comentadas anteriormente. En la grafica de la figura 65 se puede observar como la amplitud de las curvas es muy inferior y el número de infectados aumenta considerablemente.

En el caso del pico de máxima infección se puede observar como supera el doble de infectados que en la simulación con el valor 3 de R0 y la duración de la enfermedad se reduce a prácticamente la mitad. Esto nos indica que las observaciones realizadas en las anteriores simulaciones son bastante acertadas.

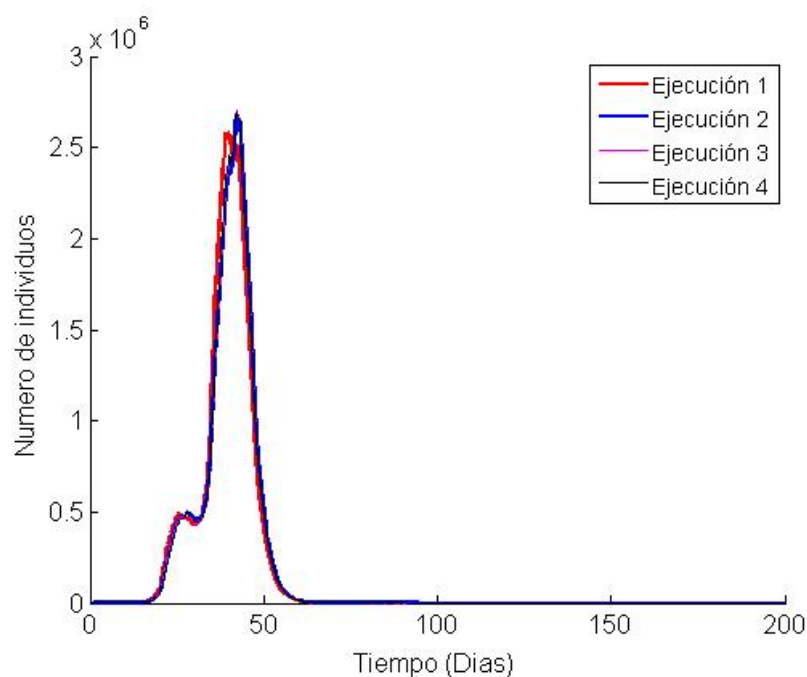


Figura 65: Epidemia con R0=3.

Como se puede observar en la tabla 54, la duración de la enfermedad se ha reducido a prácticamente la mitad al duplicar el valor de R0. De esto se puede concluir que el valor de R0 es inversamente proporcional al tiempo que dura la infección cuando este valor es mayor que 1.

También puede observarse que el número máximo de infectados en un mismo instante de tiempo es muy superior al del resto de simulaciones. En todas las simulaciones anteriores el porcentaje con respecto al total de infectados era de un 10% con el valor de R0 a 1.5 y de un 9% con el valor de R0 a 1.3. Sin embargo en esta simulación el pico de máxima infección representa

un 16% del total de infectados. Esto nos indica que este porcentaje aumenta también conforme se aumenta el valor de R_0 .

El motivo de este efecto sobre las simulaciones es debido a la rápida propagación. Conforme se aumenta el valor de R_0 la enfermedad se propaga mas rápido y los contagios se concentran cada vez en mayor medida entorno a un instante de tiempo. Esto implica que este instante de tiempo tenga un mayor número de infectados, ya que, la rápida propagación impide que se supere la enfermedad por los individuos infectados en instantes ligeramente anteriores superen al enfermedad antes de alcanzar el máximo.

Ejecución	Máximo infectados	Duración	Total Infectados	Fallecidos
Primera ejecución	2582641	64.59 días	16606451	25495
Segunda ejecución	2679246	64.54 días	16627093	24985
Tercera ejecución	2704899	64.62 días	16621825	25234
Cuarta ejecución	2698998	64.22 días	16621274	24827

Tabla 54: Resumen evaluación con $R_0=3$

Observando los resultados en el visualizador puede verse que la propagación es mucho mas rápida dándose el máximo local del foco inicial, en este caso Madrid, en un instante de tiempo muy inferior. En las simulaciones este instante de tiempo se produce entorno al día 41 mientras que en estas simulaciones el día que alcanza la máxima infección de Madrid se da entorno al día 20. En la figura 66 se observa Madrid en el máximo nivel de infección en el día 22.



Figura 66: Día 22 simulación $R_0=3$.

Posteriormente la enfermedad se propaga con gran facilidad al resto de ciudades y en torno a 5 días la enfermedad se encuentra propagada a prácticamente todas las ciudades contempladas en la simulación. Esto es debido a que con este valor de R_0 el número de infectados en Madrid es muy superior al del resto de simulaciones y, debido a esto, la

probabilidad de que un individuo infectado viaje desde Madrid a otra ciudad es muy superior. Por este motivo la propagación al resto de ciudades es prácticamente simultánea y se da un punto máximo mucho mayor al resto de simulaciones.

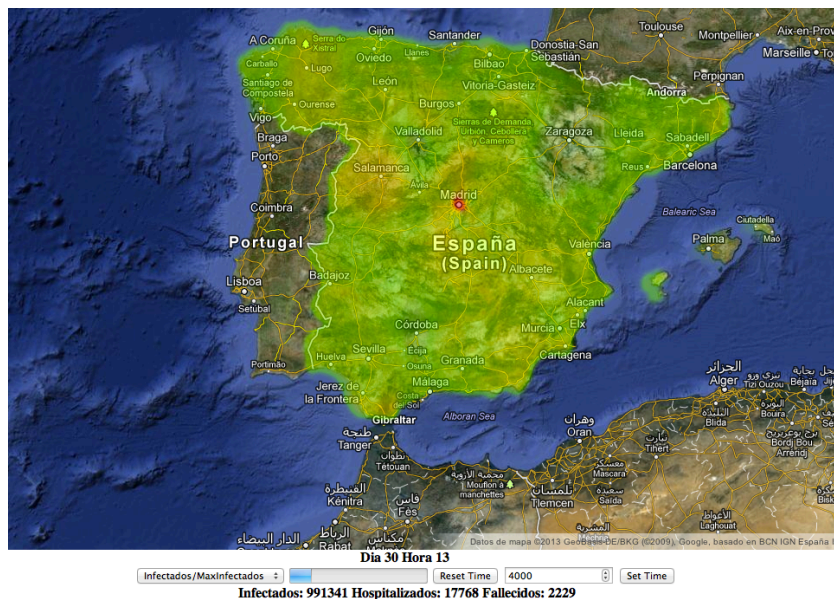


Figura 67: Día 30 simulación $R_0=3$.

Finalmente la enfermedad se supera rápidamente en toda España después de la rápida propagación dando una bajada muy rápida en el número de infectados después de superar la enfermedad.

$$R_0 = 0.5$$

En estas simulaciones el valor de R_0 es de 0.5, es decir, que en promedio, cada dos infectados transmite la enfermedad a un único individuo. En la figura 68, puede observarse como con este valor, como era de prever, la enfermedad es incapaz de desarrollarse y propagarse por el país, sino que, desaparece rápidamente e infecta a un escaso número de individuos.

Debido al bajo promedio de infectados por infectado inicial la enfermedad tiende a reducir el número de infectados a la mitad por cada cambio de generación del virus y por ello la enfermedad nunca podría desarrollarse con un valor de R_0 tan bajo. La diferencia entre las distintas simulaciones dependen únicamente del comportamiento estocástico del simulador.

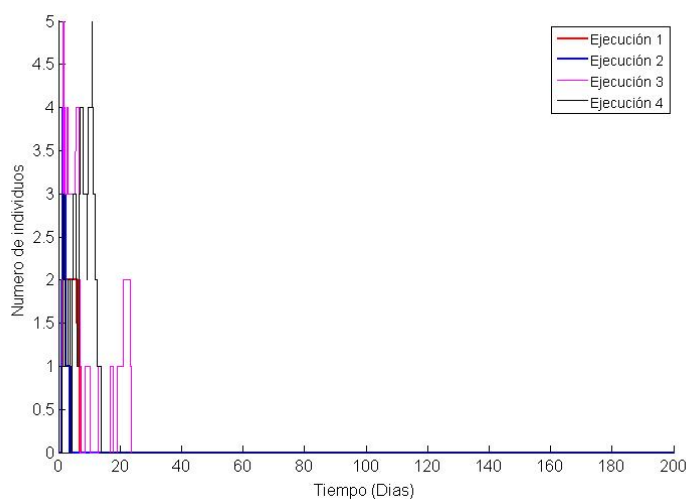


Figura 68: Epidemia con R0=0.5.

R0 = 1

Como se ha podido comprobar en las anteriores simulaciones la enfermedad se propaga con mayor velocidad y a un mayor número de individuos conforme el valor de R0 aumenta. También se ha comprobado que si este valor es menor que 1 la enfermedad es incapaz de desarrollarse y si es mayor la enfermedad es capaz de propagarse. Por este motivo se ha realizado una última prueba con el valor de R0 igual a 1, es decir cada individuo infectado transmite en promedio la enfermedad a 1 individuo lo que indica que el número de infectados en promedio deberá ser prácticamente constante a lo largo de la simulación.

De esta manera se busca comprobar el comportamiento de la propagación de la enfermedad en el punto intermedio en el que siendo mayor la enfermedad es capaz de desarrollarse y siendo inferior no es capaz de hacerlo.

Como se puede observar en la figura 69, la probabilidad de que la enfermedad sea capaz de propagarse es muy limitada y depende en gran medida de las conexiones existentes en los individuos infectados inicialmente y de la aleatoriedad del sistema.

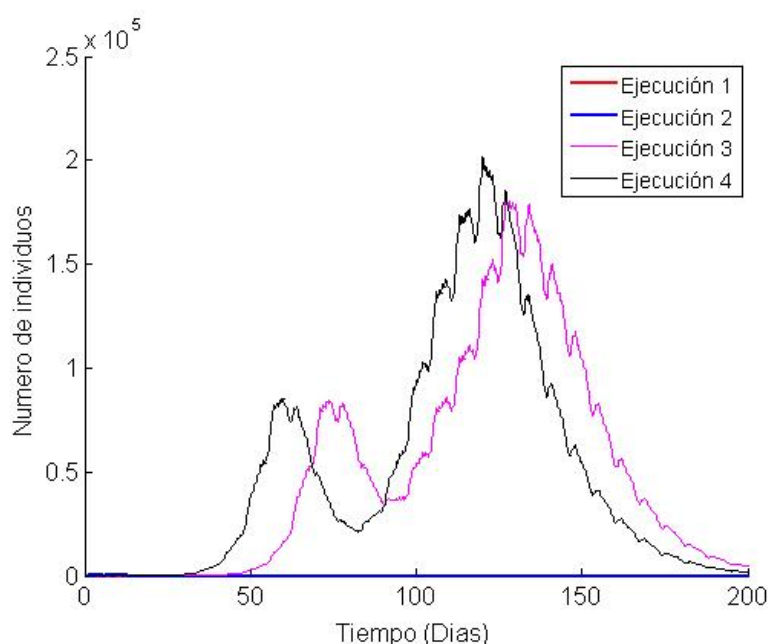


Figura 69: Epidemia con $R_0=1$.

Si la enfermedad consigue propagarse, el tiempo que se mantiene activa la enfermedad es muy superior al resto de las simulaciones, hasta el punto que como se puede observar en la tercera y cuarta ejecución la enfermedad no ha desaparecido, sino que acaba posteriormente al periodo de tiempo establecido para la simulación.

Otra de las características resultantes en la propagación con este valor de R_0 es que la enfermedad se propaga dándose gran cantidad de picos y sin alcanzar valores de número de infectados en un mismo instante de tiempo demasiado altos. Esto es debido a que el valor de R_0 a 1 implica que la enfermedad se propaga con bastante dificultad y tiende a mantenerse en relativa estabilidad en cuanto al número de infectados.

La enfermedad al propagarse en un promedio de 1 infectado por generación solo es capaz de propagarse debido a que no solo contagien aquellos individuos infectados si no que también pueden contagiar individuos en etapas de incubación de la enfermedad y enfermos latentes. Esto permite que un infectado se mantenga cierto tiempo siendo contagioso y de lugar a que la enfermedad consiga propagarse.

Ejecución	Máximo infectados	Duración	Total Infectados	Fallecidos
Primera ejecución	4	14.62 días	9	0
Segunda ejecución	4	11.54 días	16	0
Tercera ejecución	182488	> 200 días	4560926	7024
Cuarta ejecución	201423	> 200 días	4588134	7084

Tabla 55: Resumen evaluación con $R_0=1$.

Como se puede observar en la tabla 55, los valores obtenidos para cada uno de los valores analizados en la tabla mantienen los valores esperados tras las conclusiones extraídas de las simulaciones anteriores. Como se puede observar el número de infectados totales es menor y el porcentaje que representa el pico máximo con respecto al número de infectados totales es

también menor, en este caso en torno al 5%. También se ha comprobado que, cuando se reproduce la enfermedad la duración es superior dando una mayor amplitud a la curva.

Por otro lado, al ser un valor que se encuentra en el límite de los valores que permiten la propagación con relativa facilidad y aquellos que no propagan la enfermedad, la enfermedad no es capaz de propagarse en todas las simulaciones.

Observando los resultados de la segunda y tercera ejecución puede observarse como el tiempo que tarda en alcanzar el máximo local de Madrid es muy superior al resto de ejecuciones realizadas a lo largo de todas las evaluaciones. También, como se observa en la figura 70, al ser un tiempo tan extenso la enfermedad se propaga a otras ciudades antes de alcanzar este máximo en Madrid.



Figura 70: Día 62 simulación $R_0=1$.

conforme avanza el tiempo en esta simulación la enfermedad se va propagando al resto de ciudades progresivamente sin infectar una gran cantidad de ciudades en el mismo tiempo. Además al propagarse tan lentamente la infección las ciudades mantienen un valor de infección cercano a su máximo durante un gran espacio de tiempo hasta llegar al máximo número de infectados simultáneamente en la ejecución.

En las siguientes figuras se puede observar como la enfermedad se propaga lentamente al resto de ciudades en diferentes tiempos y una vez alcanzan su máximo lo mantienen a lo largo de un número de días bastante alto en comparación al resto de simulaciones anteriores. Esto puede observarse claramente en la zona situada entre Burgos y Valladolid.

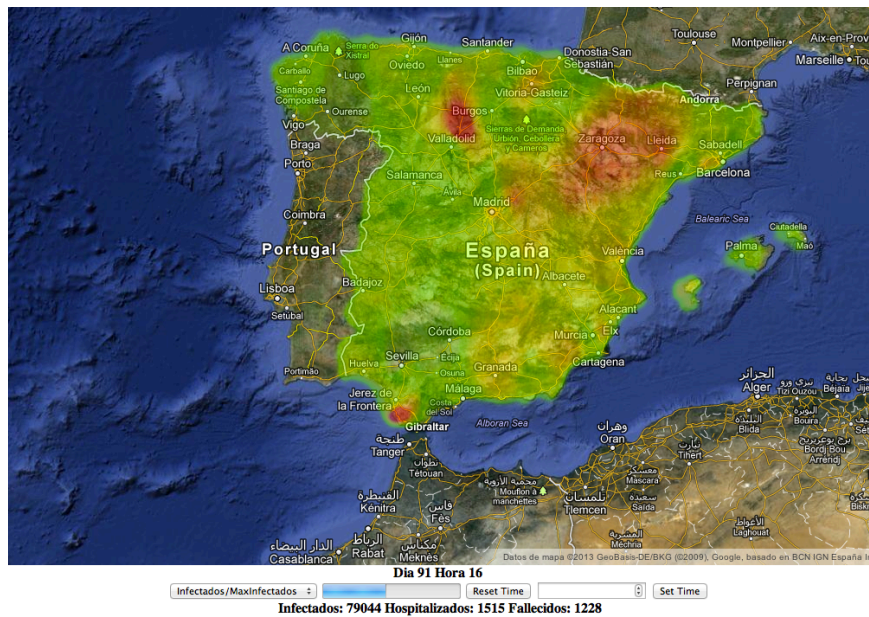


Figura 71: Día 91 simulación $R_0=1$.

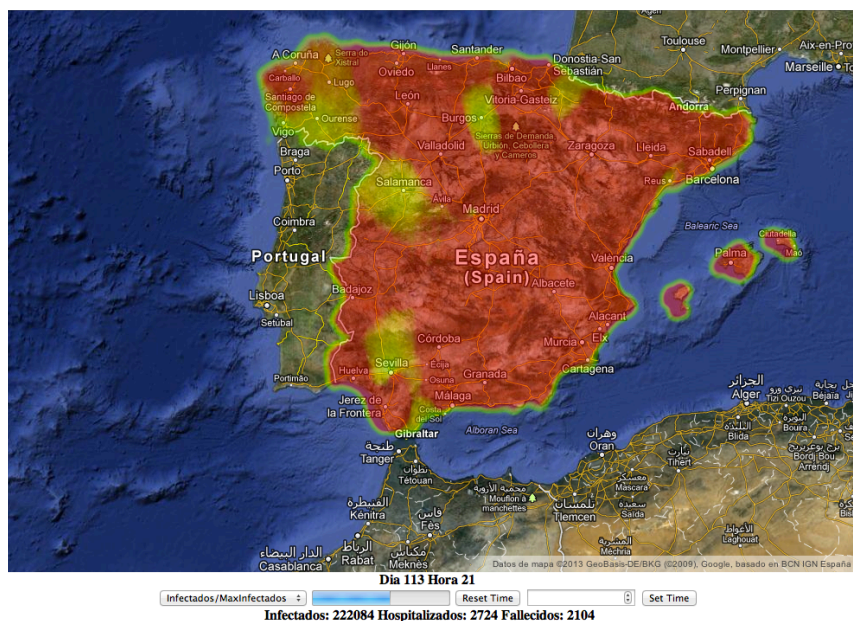


Figura 72: Día 113 simulación $R_0=1$.

Posteriormente la enfermedad va desapareciendo muy lentamente de las ciudades infectadas.

Una vez realizadas todas las simulaciones con los diferentes valores de R_0 se ha analizado como influye estos cambios en el número de infectados. Para ello se ha realizado una gráfica que muestra los valores medios del número de infectados, en las ejecuciones que propagan la enfermedad, con respecto al valor de R_0 .

Como se observa en la gráfica 73, el valor de R_0 implica un comportamiento logarítmico en cuanto al número de infectados totales. De aquí puede predecirse que con valores mayores de R_0 mayores el aumento empieza a ser menor progresivamente hasta alcanzar en valores de R_0 suficientemente altos el valor del total de la población. La siguiente gráfica se ha generado utilizando los valores obtenidos a lo largo de esta sección añadiendo el valor de R_0 , utilizado en el resto de secciones, de 1.3.

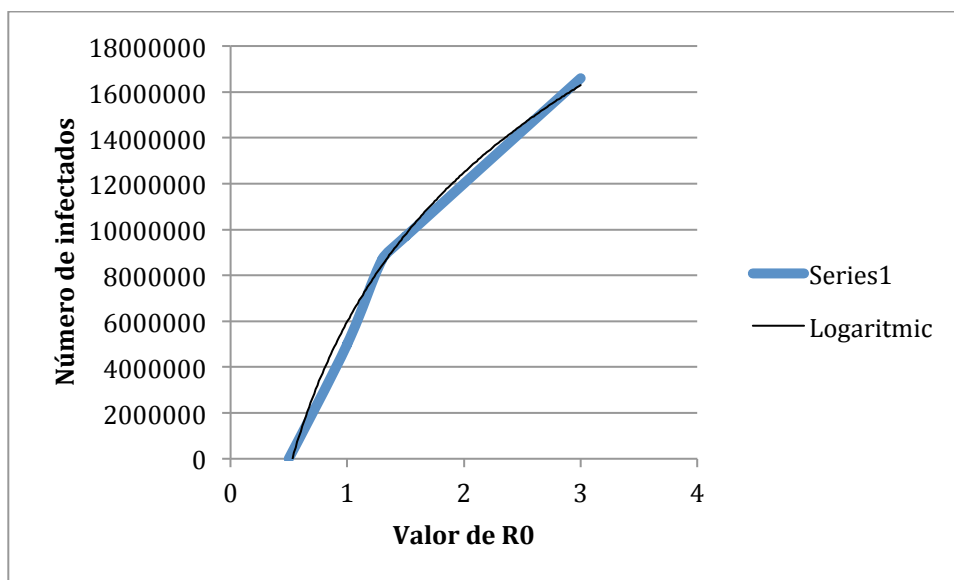


Figura 73: Gráfica infectados totales respecto a R_0 .

6.5. Evaluación con focos compuestos.

En esta sección se busca evaluar el comportamiento de la enfermedad cuando los infectados iniciales se encuentran dispersados en diferentes ciudades. Para ello se han realizado cuatro ejecuciones partiendo de distintos números de infectados divididos en las siguientes ciudades: Madrid, Barcelona, Leganés y A Coruña.

Para realizar esta evaluación sobre la propagación de la enfermedad con varios focos se realizarán dos bloques de cuatro ejecuciones idénticas cada una. El primero de estos bloques se partirá de 1 infectado en cada ciudad y el segundo con 3 por ciudad.

Evaluación con 1 infectado por ciudad.

En esta evaluación se parte con un individuo infectado inicialmente en cada una de las ciudades que componen el foco inicial de la infección. Con esta evaluación se busca comprobar como se propaga la enfermedad con un número pequeño de infectados iniciales repartidos en distintas ciudades.

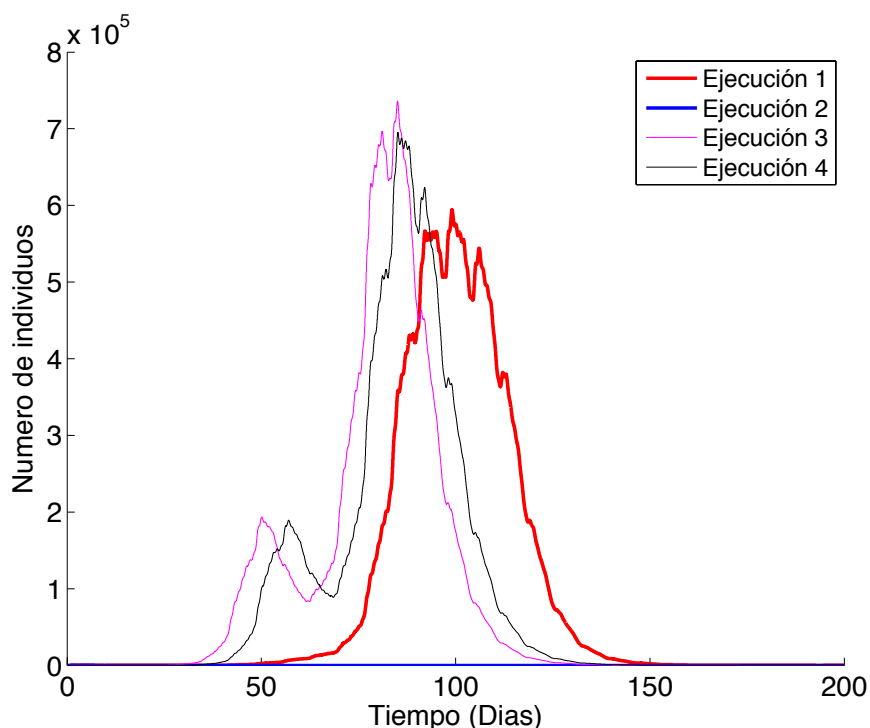


Figura 74: Epidemia con 1 infectado en cada foco.

En la figura 74 se puede observar las grandes variaciones entre las ejecuciones dependiendo de que foco o focos son capaces de reproducir la enfermedad partiendo de un número tan limitado de infectados iniciales.

Al tratarse de escenarios en los cuales el número de infectados es uno por cada foco, la propagación de la enfermedad dentro de estas ciudades se hace bastante difícil lo cual implica que estos focos inicien su enfermedad en tiempos distintos o que no sean capaces de desarrollar la enfermedad.

En la segunda ejecución puede observarse que la enfermedad no es capaz de desarrollarse en ninguno de los focos lo que implica que no se produzca la epidemia en el resto del país.

De igual manera se producen las variantes en la propagación de la enfermedad dependen de aquellos focos que son capaces de desarrollar la enfermedad.

Ejecución	Máximo infectados	Duración	Total Infectados	Fallecidos
Primera ejecución	593849	134,92 días	8718068	13428
Segunda ejecución	3	0,3 días	11	0
Tercera ejecución	736080	130,92 días	8713908	13148
Cuarta ejecución	695414	129,86 días	8714673	13388

Tabla 56: Resumen de ejecuciones con 1 infectado por foco.

A continuación, haciendo uso de la herramienta de visualización desarrollada se busca analizar las diferentes ejecuciones con el fin de observar el comportamiento en la propagación de la enfermedad en las diferentes ejecuciones.

En la primera ejecución, como se observa en la figura 75, la enfermedad se desarrolla únicamente en A Coruña dando un resultado similar al obtenido simulando con esta ciudad como foco inicial.



Como se muestra en la figura 76, la enfermedad se propaga a las distintas ciudades en un intervalo de tiempo espaciado, lo que implica que el punto en que se alcanza el mayor número de infectados es bastante inferior al no darse simultáneamente en todas las ciudades.

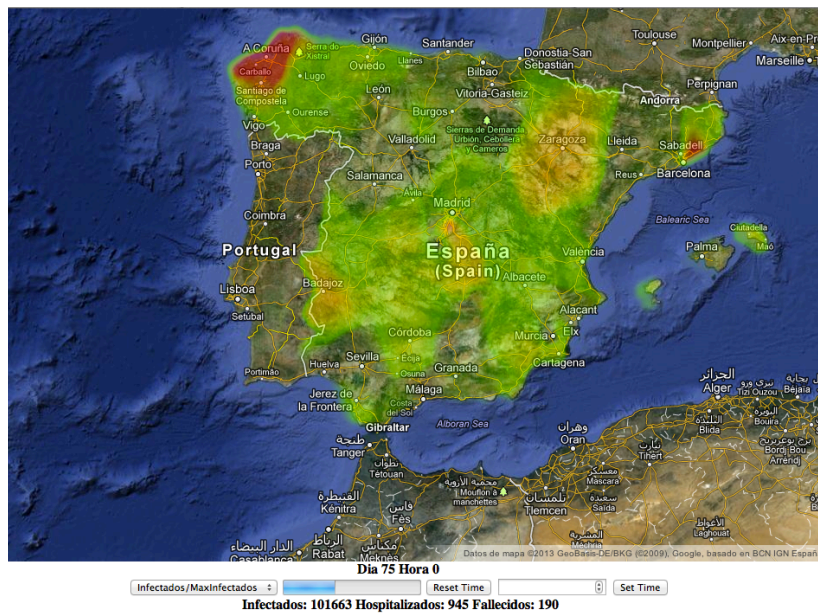


Figura 76: Día 75 primera simulación con 1 infectado en varios focos.

Por este mismo motivo, el tiempo en que se mantiene activa la enfermedad es superior al del resto de simulaciones donde se propaga a todas las ciudades en un intervalo de tiempo menor.

En la segunda simulación se da la situación de que la enfermedad no es capaz de desarrollarse en ninguna de las ciudades al existir un número tan pequeño de infectados iniciales en cada ciudad. Debido a esto la enfermedad no se propaga en ninguno de los focos y por este motivo no llega a salir de ellos.

La tercera ejecución muestra un número máximo de infectados mayor y se propaga en un intervalo de tiempo menor. Como se puede observar en la figura 77, esto se debe a que la enfermedad consigue propagarse simultáneamente en dos focos: A Coruña y Madrid.



Figura 77: Día 45 tercera simulación con 1 infectado en varios focos.

Esto implica que ambos focos propagan la enfermedad al resto de ciudades y se infectan un mayor número de ciudades en el mismo intervalo de tiempo. En la figura 78, puede observarse como se propaga la enfermedad de una forma prácticamente homogénea a todo el país, lo que supone que la mayoría de ciudades alcancen su pico máximo en un mismo instante de tiempo y por tanto el máximo número de infectados sea mayor que en el resto de ejecuciones.

Por este motivo también el tiempo que permanece activa la enfermedad se reduce en comparación al resto de simulaciones.

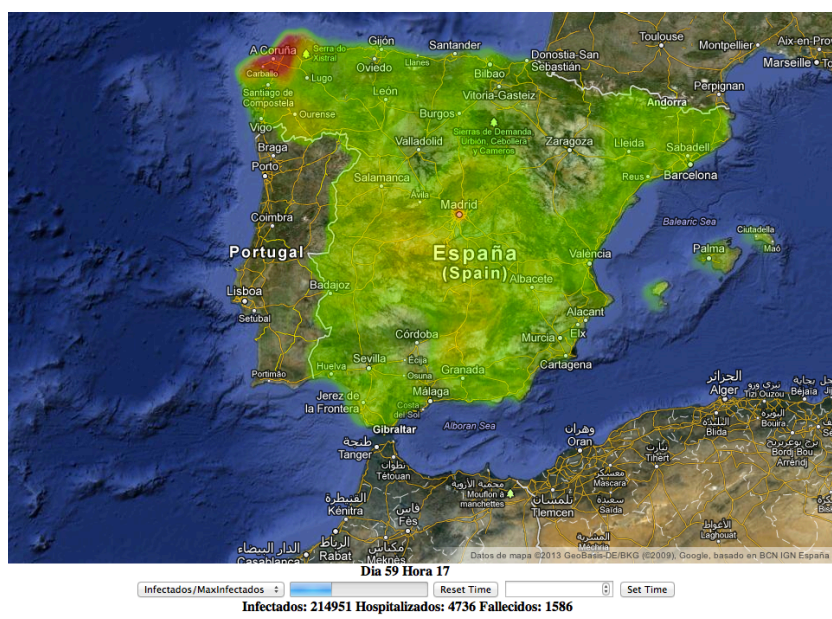


Figura 78: Día 59 tercera simulación con 1 infectado en varios focos.

En la última simulación realizada con esta configuración de infectados iniciales se da un resultado similar a la ejecución realizada con el foco inicial en Madrid debido a que es en esta ciudad en la que la enfermedad consigue propagarse. En la figura 79 se muestra como se desarrolla la enfermedad únicamente en Madrid.



Figura 79: Día 51 cuarta simulación con 1 infectado en varios focos.

Una vez realizado el análisis de estas simulaciones puede concluirse que la propagación depende del número de infectados iniciales en una única ciudad, debido a que con un número de infectados demasiado bajo puede no llegar a desarrollarse o desarrollarse más lentamente.

Evaluación con 4 infectados por ciudad.

En esta segunda evaluación se han introducido 4 infectados iniciales en cada ciudad con el fin de comprobar como se propaga la enfermedad cuando el número de infectados iniciales utilizado en las evaluaciones con infectados en una única ciudad se encuentra en varias ciudades simultáneamente.

Observando las gráficas de la figura 80 se puede observar un primer pico en el número de infectados que varía en su número dependiendo del número de ciudades que desarrollan la enfermedad en ese momento.

Si comparamos el tamaño de este pico con los que se obtenían de resultado de simular la infección en una sola ciudad se puede observar que los picos más altos coinciden con el número acumulado de máximo desarrollo de la enfermedad en todas las ciudades iniciales. Esto indica que la enfermedad, al partir de un número de 4 infectados es capaz de desarrollarse con facilidad dentro de la ciudad dando lugar a que en estas simulaciones tienda a propagarse la enfermedad en las 4 ciudades prácticamente de forma simultánea.

Por otro lado se puede apreciar también que el número de infectados totales permanece prácticamente constante, salvo pequeñas variaciones dependientes del factor de aleatoriedad del sistema, en todas las ejecuciones realizadas con el mismo valor de R_0 , 1.3.

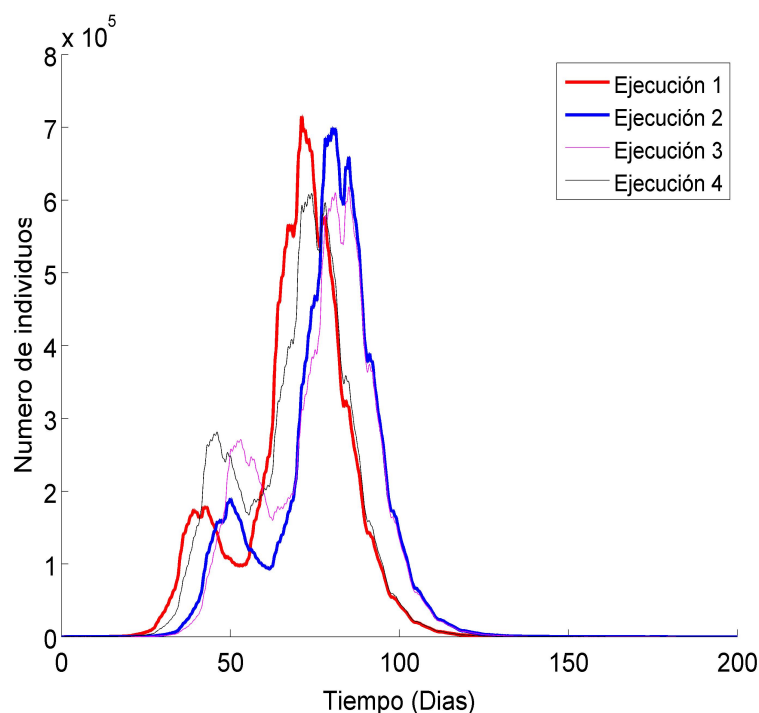


Figura 80: Epidemia con 4 infectados en cada foco.

Sin embargo, si se centra la atención en el valor máximo del número de infectados se observa que el valor máximo es ligeramente inferior al obtenido en las simulaciones con un único foco. Esto es debido a que al infectarse las 4 ciudades antes que el resto de ciudades, cuando se produce la propagación, estas ciudades han superado la enfermedad. Debido a que las ciudades donde se inicia la enfermedad tienen una gran población esto supone que a la hora de propagarse un gran porcentaje de la población ya ha superado la enfermedad.

Tabla 57: Resumen de ejecuciones con 4 infectados por foco.

En la primera ejecución, como se observa en la figura 81, la enfermedad se desarrolla tanto en Madrid como en A Coruña. Esto da lugar al primer pico de la gráfica, en torno al día 40, en el momento que ambas ciudades alcanzan su máximo local de la enfermedad. Posteriormente la enfermedad se propaga a un gran número de ciudades lo que dará lugar al segundo pico de la infección.



Debido a que la enfermedad no se propaga a todas las ciudades por igual sino que se da en periodos distintos, el periodo de tiempo es ligeramente superior al resto de ejecuciones. Por otro lado el número máximo de infectados también es superior a las 2 ultimas ejecuciones. Esto es debido a que Barcelona, que es una ciudad bastante poblada, no se infecta al inicio sino que se infecta junto a la mayoría de ciudades del país. Esta diversidad en el momento en que se propaga la enfermedad puede observarse en la figura 82 donde se muestra la diversidad en la gran variedad de intensidades de color del país.

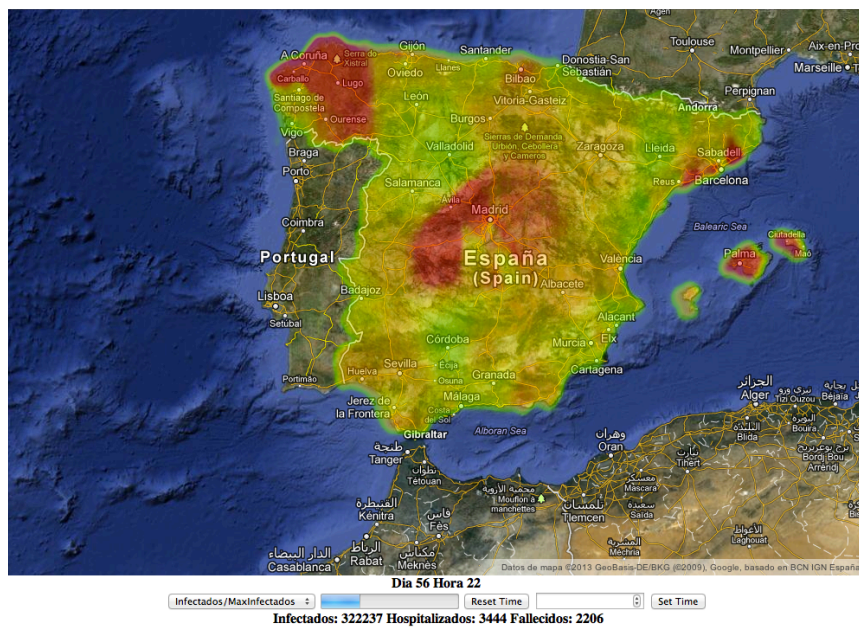


Figura 82: Día 56 primera simulación con 4 infectados en varios focos.

En cuanto a la segunda ejecución la propagación de la enfermedad se da igual que en la primera ejecución salvo por una diferencia de tiempo en el momento en que se inicia la enfermedad. Como se puede observar en la figura 83, la enfermedad se propaga en las ciudades de Madrid y A Coruña con una diferencia de 10 días con la primera ejecución. Esto implica que la curva dibujada en la gráfica para esta ejecución sea muy similar a la de la primera con un desplazamiento de 10 días.



Figura 83: Día 47 segunda simulación con 4 infectados en varios focos.

En cuanto a la tercera y cuarta ejecución el comportamiento de la propagación de la enfermedad es bastante distinto a las dos anteriores simulaciones. En estos casos la enfermedad se propaga tanto en Madrid como en Barcelona. En la figura 84 se muestra el estado de la enfermedad en la tercera simulación donde se puede observar Madrid y Barcelona infectadas.



Figura 84: Día 48 tercera simulación con 4 infectados en varios focos.

Como se puede observar existen diferencia en el tiempo de propagación de la enfermedad. Esto es debido a que en la tercera simulación, como se ha podido observar en la diferencia de la intensidad de la enfermedad en ambas ciudades es diferente. Esto indica que el tiempo que ha tardado en desarrollarse la enfermedad en Barcelona es ligeramente superior que en Madrid. Por este motivo el primer pico de la enfermedad es algo inferior debido a que la enfermedad empieza a desaparecer en Madrid cuando Barcelona alcanza su pico máximo.

Si comparamos con la cuarta simulación, la enfermedad en esta simulación si se desarrolla en ambas ciudades prácticamente de forma simultánea, como se puede observar en la figura 85, lo que da un primer pico algo más alto. Además, se observa que el desplazamiento, igual que ocurría en las dos primeras simulaciones, viene debido al tiempo que tarda en desarrollarse inicialmente la enfermedad.



Figura 85: Día 34 cuarta simulación con 4 infectados en varios focos.

En ambas simulaciones se da el pico máximo al propagarse la enfermedad al resto de ciudades, aunque a diferencia de las dos primeras simulaciones, se trata de un valor máximo algo inferior. Esto es debido a que se infectan las dos ciudades más pobladas del país Madrid y Barcelona antes que el resto de ciudades y por tanto, la enfermedad está en proceso de desaparecer cuando se alcanza el pico máximo.

Por otro lado el tiempo que permanece activa la enfermedad es ligeramente inferior debido a una propagación más rápida que en los dos primeros casos, debido a que tanto Barcelona como Madrid son ciudades muy interconectadas y propagan la enfermedad al resto de ciudades casi simultáneamente.

7. Planificación.

En esta sección se describe la planificación seguida en el desarrollo del proyecto. Se distinguen 4 fases en el desarrollo: Análisis, Diseño, Implementación y Pruebas.

Se ha estimado para el tiempo de desarrollo del sistema un total de 91 días desde el 22 de enero de 2013 hasta el 1 de junio de 2013. En la tabla 58 se muestra el desglose de los días asignados para cada una de las fases del desarrollo del proyecto:

Fase	Duración (días)	Fecha de inicio	Fecha de fin
Análisis	13	22/01/2013	06/02/2013
Diseño	17	07/02/2013	05/03/2013
Implementación	31	06/03/2013	19/04/2013
Pruebas	18	22/04/2013	16/05/2013
Documentación	78	06/02/2013	1/06/2013

Tabla 58: Planificación.

A continuación, en la figura 86, se muestra el diagrama de Gantt para la planificación del proyecto con el fin de mostrar de una forma más visual la división del trabajo en las diferentes etapas a lo largo del proyecto.

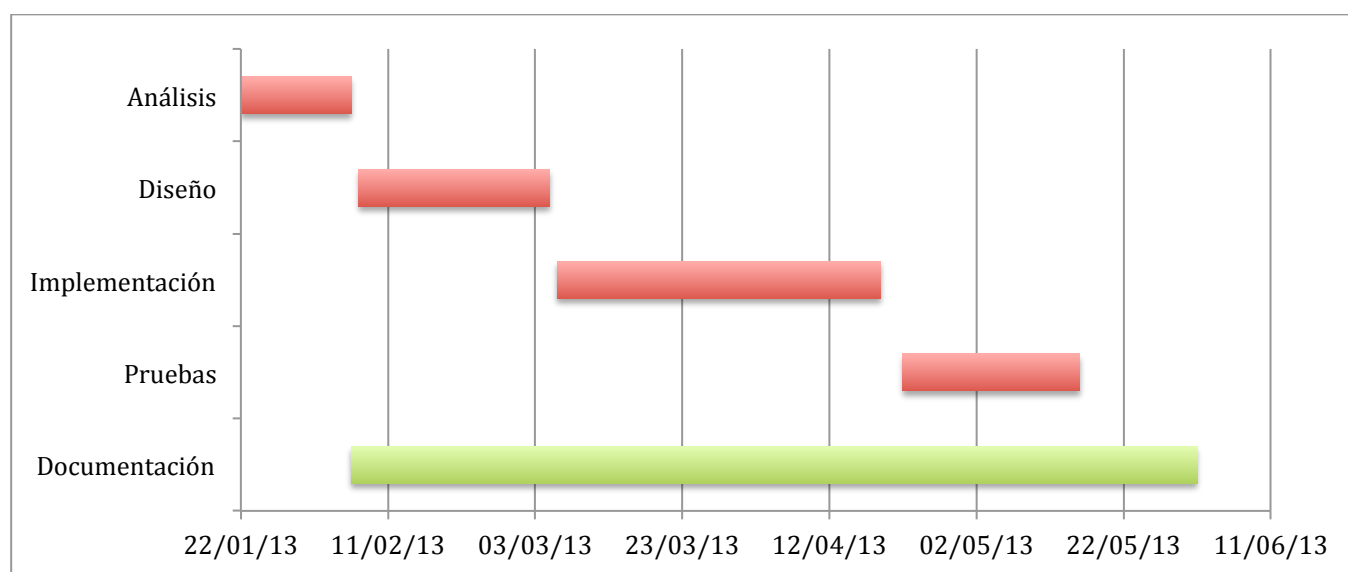


Figura 86: Diagrama de Gantt.

8. Presupuesto.

En esta sección se describe una primera estimación de los costes correspondientes a este proyecto.

Recursos humanos.

Para el desarrollo de este proyecto será necesario un equipo de 3 personas. Un jefe de proyecto, un programador y un analista. A continuación se muestra una tabla con el desglose de los salarios de los componentes del equipo.

	Jefe de Proyecto	Analista	Programador
Bruto/mes	3.700,00 €	3.000,00 €	2.300,00 €
Neto/mes	2.331,00 €	1.890,00 €	1.656,00 €
Bruto/año	51.800,00 €	42.000,00 €	32.200,00 €
Neto/año	32.634,00 €	26.460,00 €	23.184,00 €

Tabla 59: Salarios recursos humanos.

Para el calculo del salario neto se calcula el tramo en que se encuentra según el salario anual de cada componente y se descuenta el IRPF. Los tramos de IRPF, definidos para el año 2013 por la Ley 35/2006 del 28 de noviembre, son los siguientes:

Salario Bruto		Retención
Desde	Hasta	
0,00 €	9.050,00 €	0%
9.051,00 €	17.360,00 €	24%
17.361,00 €	32.360,00 €	28%
32.361,00 €	52.360,00 €	37%
52.361,00 €		43%

Tabla 60: Presupuesto IRPF.

Finalmente, tras las reducciones sobre el salario por el IRPF se calcula el coste total de los salarios dentro del proyecto multiplicando el salario bruto por los 4 meses planificados.

	Jefe de proyecto	Analista	Programador
Total	14.800 €	12.000 €	9.200 €

Tabla 61: Presupuesto recursos humanos.

De estos cálculos se extrae que el coste total de recursos humanos para este proyecto asciende a un total de **36.000 €**.

Equipamiento

A continuación se detallan los costes del equipamiento necesario para llevar a cabo el proyecto:

	Cantidad	Precio	Total/mes	Total
Sitio de trabajo	1,00	500,00 €	500,00 €	2.000,00 €
Ordenadores	3,00	700,00 €	21,88€	87,50 €
Mesas	2,00	125,00 €	93,75 €	375,00 €
Sillas	3,00	75,00 €	56,25 €	225,00 €
Conexión a internet	1,00	45,00 €	45,00 €	180,00 €
Red Eléctrica	1,00	53,80 €	53,80 €	215,20 €
Memorias USB	3,00	15,00 €	15,00 €	45,00 €
Cluster	1,00	4.500,00 €	46,88 €	187,50 €

Tabla 62: Presupuesto equipamiento.

Según lo descrito en Real Decreto 1777/2004 se establece que los elementos de tratamiento de información tienen un plazo de amortización de 8 años. Debido a esto el coste de los ordenadores y del clúster se ha dividido entre los 96 meses que establece el plazo de amortización.

Para distribuir los costes de los equipos de un único pago a lo largo de los meses se ha dividido el coste total entre el número de meses de desarrollo estimados en la planificación.

Una vez realizados estos cálculos se procede a calcular los costes totales del equipamiento multiplicando el coste mensual por los meses estimados.

Tras estos cálculos el coste del equipamiento asciende a un total de **3.315,20 €**.

Sumando los costes de equipamiento con los de recursos humanos el coste total del proyecto asciende a **39.315,20 €**.

Coste de Software

Para llevar a cabo el desarrollo de este proyecto serán necesarias licencias de trabajo para los programas a utilizar en el proyecto. En la siguiente tabla se muestra el desglose de los costes de licencias para cada uno de los programas.

Software	Unidades	Precio
Microsoft Office	1,00	200 €
MatLab	1,00	6.000 €

Tabla 63: Presupuesto de Software.

El coste total de las licencias necesarias es de **6.200 €**. Añadiendo estos costes el coste total del proyecto asciende a **45.515,20 €**.

Coste de riesgo

Con el fin de asegurar la solvencia del proyecto ante posibles imprevistos derivados de la mala gestión económica se aplica una tasa del 10% sobre el coste para subsanar los posibles imprevistos.

El total del coste de riesgo es de **4.551,52 €**.

Beneficios

Una vez calculado el coste se establece que se debe asumir un beneficio del 20% sobre el coste del proyecto. Esto es un total de **9.103,04 €** de beneficio.

Incluyendo estos costes el total del proyecto asciende a **52.969,76€**.

Impuestos

A este coste hay que añadir la tasa, estipulada por la Ley 2/2012, de 29 de Junio del 2012 en su capítulo 2 sección de impuestos indirectos, del 21 % correspondiente al Impuesto sobre el Valor Añadido (I.V.A.).

El coste total del proyecto incluyendo los impuestos es de **64.093,41 €**.

En la tabla 64 se muestra el resumen de los costes de cada uno de los conceptos detallados anteriormente.

Concepto	Coste
Recursos humanos	36.000,00 €
Equipamiento	3.315,20 €
Software	6.200,00 €
Coste de Riesgo	4.551,52 €
Beneficios	9.103,04 €
Impuestos	11.123,65 €
TOTAL	64.093,41 €

Tabla 64: Resumen del presupuesto.

9. Conclusions and future work.

This section describes the conclusions drawn from the achievement of the objectives set out in this paper. Also included are personal conclusions obtained after complete the development of the project and a guidelines to be followed for future work on the same project.

9.1 *General conclusions.*

After reaching the objectives described in section 1.2 about the objectives the project have been established the following conclusions.

Reaching the first objective about the implementation of a graph sampling algorithm to generate the simulated population connections have been develop new connectivity graphs whose structure is more realistic than the algorithm that EpiGraph implements initially.

Moreover, relative to second objective, which was intended to develop an application to view the results of the simulations performed in epigraph on a map, has been successfully developed a tool in HTML and JavaScript able to perform this task with a high level of quality. It has also been concluded that the selection of the technology to develop the application has been successful to be able to develop the application with great quality. This has been possible due to the API that provides Google maps, which can display and apply different layers on the map with ease.

After the achievement of the third objective established for this project, which was performing an analysis of the impact of the graphs structure in the spread of the epidemic, it has been concluded that the impact that represent the different structures on the epidemic is quite important. By using new graph generated by the algorithm implemented in the development of this project, more realistic results have been achieved in the simulations. However, it has been found a great difference in the computation time needed to generate the graphs using this new algorithm compared to scaling algorithm, which means that too large populations computation cost may be an important limiting factor. This exhaustive analysis has taken one hundred and seven hours of CPU time to allow comparing accurately the difference on computing the cost of applying both algorithms with different population sizes.

Finally, from the achievement of fourth object, which the objective was determined as an analysis of the results of simulations under different scenarios, one can conclude the great impact of variations in the parameters, which define the disease such as R_0 and the different foci in which the disease begins on the forecasts obtained from the simulation. Of this as well is drawn the conclusion that for a good forecast for spread of a disease, it is necessary to enter the parameters which define the disease with great accuracy and it is also very important to determine the point of onset of the disease.

9.2 *Personal conclusions.*

This project has been concluded, from a personal point of view, two fundamental ideas.

On one hand the issue of epidemiology and forecast is quite interesting and has great importance in society. These tools, which simulate the diseases for forecasting, are useful when dealing with a new disease. For this reason it is so important to find ways to make more accurate forecasts. For this reason these simulators use graphs based on social networks to

simulate the connections between individuals and thus provide a model on which simulate the disease as close to reality as possible.

On the other hand, the conclusion drawn from the visual interface development with HTML and JavaScript, is that, despite the lower performance of these technologies compared to native application development can get great results as it has been proved. An important part of success on the development of this application is the use of GoogleMaps library, which offers a lot of resources to work on maps. From the use of this library can conclude that is a quite powerful API allows an easy implementation of applications that work over maps.

Finally, indicate that it has reached a great feeling of satisfaction at having completed every one from the objectives with a high level of quality in the products deriving from the process of development work.

9.3 Future research.

Along the development of the project have been raised potential improvements towards the future as well as new analysis that could be interesting. This section describes some of these ideas.

In first place, an important analysis was study the effect of the use of disease containment strategies comparing simulations with different threshold of use these strategies. Thus could be seen how it reflects the use of these strategies on the number of infected as well as in the duration of the disease and propagation speed.

Moreover, a useful implement for EpiGraph could be introduces climatological data of simulated city to improve the accuracy of forecast. Because the disease spreads differently depending on the weather, using past climatic data of these cities can be simulated the disease using these climatic data to modify the properties of the spread of the disease. Following this line could be included in the visual interface a new layer or a new instance of the map that shows climatic data used in the simulation. Thus could studied the propagation of the disease and observe the climate of each city simultaneously giving an overview of the results of the simulation.

Once introduced this improvement, the next step would be to conduct a comprehensive study of the effect of applying this improvement in simulation results comparing these new results with the results without using climatic data.

10. Referencias.

- [1] Long Jin, Yang Chen, Pan Hui, Cong Ding, Tianyi Wang, Athanasios V. Vasilakos, Beising Deng and Xing Li. "Albatross Sampling: Robust and Effective Hybrid Vertex Sampling for Social Graphs". Proceeding, HotPlanet '11 Proceedings of the 3rd ACM international workshop on MobiArch, Pages 11-16, 2011. http://delivery.acm.org/10.1145/2010000/2000178/p11-jin.pdf?ip=163.117.91.209&acc=ACTIVE%20SERVICE&key=C2716FEBFA981EF1576EFC80FDC4C38F4CB2162AFE87DD7B&CFID=310831218&CFTOKEN=15149277&_acm_=1365585223_6631d21065cd0e741215f3d932c0f084
- [2] Yuming Mai. "Uniform Sampling of the Facebook Social Network Using the Metropolis Hasting Random Walk". ACM Transactions on Applied Perception, Vol. 2, No. 3, Article 1, 2010. http://davinci.newcs.uwindsor.ca/richard/cs510/survey_mai.pdf
- [3] Min-Ling Zhang and Zhi-Hua Zhou. "ML-KNN: A lazy learning approach to multi-label learning". Pattern Recognition, Vol. 40, Issue 7, pages 2038-2048, 2007. <http://www.sciencedirect.com/science/article/pii/S0031320307000027>
- [4] Google API v3 Reference. <https://developers.google.com/maps/documentation/javascript/reference>
- [5] JavaScript Reference. <http://www.w3schools.com/jsref/>
- [6] Survey of Sparse Matrix Storage Formats. <http://netlib.org/utk/papers/templates/node90.html>
- [7] An introduction to the Message Passing Interface (MPI) using C. <http://condor.cc.ku.edu/~grobe/docs/intro-MPI-C.shtml>
- [8] ArcGIS. <http://www.arcgis.com/about/>
- [9] WebKit View Reference. <https://developer.apple.com/library/mac/#documentation/Cocoa/Conceptual/DisplayWebContent/DisplayWebContent.html>
- [10] L. Lovász. "Random Walk on Graphs: A survey". Combinatorics, Paul Erdős is Eighty, Vol. 2, 1993.
- [11] K. Bisset, J. Chen, X. Feng, V. Kumar, and M. Marathe. "Epifast: a fast algorithm for large scale realistic epidemic simulations on distributed memory systems". Proceedings of the 23rd international conference on Supercomputing. ACM, 2009.
- [12] D. Chao, M. Halloran, V. Obenchain, and I. Longini. "FluTE, a publicly available stochastic influenza epidemic simulation model". PLoS computational biology, 6(1):e1000656, 2010.
- [13] C. L. Barrett, K. R. Bisset, S. G. Eubank, X. Feng, and M. V. Marathe. "EpiSimdemics: an efficient algorithm for simulating the spread of infectious disease over large realistic social networks". Proceedings of the 2008 ACM/IEEE conference on Supercomputing, page 37. IEEE Press, 2008.
- [14] Teri Jhonson. "Mathematical Modeling of Disease: Susceptible-Infected-Recovered (SIR)



Model". University of Minnesota, 2009.

- [15] Fred Brauer. "Compartamental models for epidemics". University of British Columbia, 2008.
- [16] Cécile Viboud et al. "Synchrony, Waves and Spatial Hierarchies in the Spread of Influenza. Science 312, 447, 2006.

Anexo I: Manual de usuario.

Contenido.

Este anexo contiene la información necesaria para el uso y configuración del visualizador de los resultados de las simulaciones de EpiGraph. En este manual se detallan los requisitos mínimos de la aplicación y las tareas a seguir para realizar una visualización.

Requisitos hardware y software.

En esta sección se describen los requisitos mínimos necesarios para la ejecución de la aplicación tanto a nivel de hardware como a nivel de software.

Requisitos hardware.

- Procesador: Intel Pentium IV 3 GHz o superior.
- Memoria RAM: 1 Gb.
- Espacio libre de disco duro: 800 Mb.
- Conexión a internet.

Requisitos software.

- Un navegador web capaz de ejecutar JavaScript y HTML. La aplicación es compatible con Safari, Chrome y Firefox.
- La aplicación MatLab.

Guía rápida de uso.

En esta sección se describe brevemente el proceso a seguir para ejecutar la visualización de una simulación de EpiGraph partiendo de los resultados de salida.

1. En primer lugar los archivos `v_stats[*].out` de cada ciudad deben ser introducidos en un único fichero de datos `v_stats0.out`. Para ello en la carpeta raíz donde se encuentran las carpetas con los ficheros de salida del simulador debe ejecutarse con matlab el script `stats.m`.

2. Tras realizar esta ejecución deben copiarse las carpetas `/nombreCiudad` donde se encuentran los ficheros `stats` en la carpeta `epiViewer/stats/`.

3. Si en la simulación no se han cambiado las ciudades por defectos debe ejecutarse el script `copy.sh` situado en la carpeta raíz de la aplicación `epiViewer`.



En caso contrario, deberá modificarse en `confCiudades.txt` el número de ciudades de la simulación introduciendo el número usado en la simulación y modificar las ubicaciones en el fichero `Ubicaciones`. El formato a seguir dentro del fichero `Ubicaciones` es el siguiente:

```
LatitudCiudad0  
LongitudCiudad0  
LatitudCiudad1  
LongitudCiudad1  
...
```

Debido a que el script solo sirve para las ciudades por defecto deberá realizarse la copia de los ficheros `vstats.out` de las ciudades a las carpetas, identificadas con el número de ciudad y en el orden en que están incluidas las ubicaciones en el fichero `Ubicaciones`, situadas en la carpeta raíz de la aplicación.

4. Una vez realizada la carga de fichero solo es necesario ejecutar haciendo doble clic sobre el archivo `Main.html` en la carpeta raíz de la aplicación.